Double hashing is computable and randomizable with universal hash functions

Jeanette P. Schmidt
Department of Computer Science
Polytechnic University
333 Jay Street
Brooklyn, NY 11201

Alan Siegel
Department of Computer Science
Courant Institute
251 Mercer Street
N.Y.C., NY 10012

Abstract

Universal hash functions that exhibit $c\log n$-wise independence are shown to give a performance in double hashing and virtually any reasonable generalization of double hashing that has an expected probe count of $\frac{1}{1-\alpha}+\epsilon$ for the insertion of the αn-th item into a table of size n, for any fixed $\alpha<1$ and $\epsilon>0$. This performance is within ϵ of optimal. These results are derived from a novel formulation that overestimates the expected probe count by underestimating the presence of partial items already inserted into the hash table, and from a sharp analysis of the underlying stochastic structures formed by colliding items.

Categories and Subject Descriptors: E.1 [**Data**]: Data Structures—*arrays*; *tables*; E.2 [**Data**]: Data Storage Representations—hash-table representations; F2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems—*sorting and searching*.

General terms: Algorithms, Theory.

Additional Key Words and Phrases: Double hashing, hashing, limited independence, open addressing, random probing, uniform hashing, universal hash functions.

The work of the first author was supported in part by NSF grant CCR-9110255 and the New York State Science and Technology Foundation Center through its Center for Advanced Technology.

The work of the second author was supported in part by ONR grant N00014-85-K-0046, NSF grants CCR-8906949, CCR-8902221 and CCR-9204202.

Summary

This paper gives the first performance bounds for classical closed hashing schemes in the case of limited randomness, and thereby provides the first randomized performance analysis of these algorithms in a model that supports programmable computation. In contrast, traditional analyses have relied upon the use of mathematically random hash functions or the assumption that the input data is completely random. Unfortunately, real data is seldom provably random, and the average program size of a random hash function is so large it exceeds the size of the database it is intended to support. The bounds for limited randomness establish near optimal randomized performance for classical double hashing when restricted to programmable hash functions. Moreover, the proof technique unifies and significantly generalizes previous results even in the case of unlimited randomness.

Let $D = (x_1, x_2, \ldots, x_{\alpha n})$ be a sequence of αn distinct search keys, for $\alpha < 1$, belonging to the universe $U = \{0, 1, \ldots, m-1\}$. The objective is to hash D into a search table of n locations without the use of pointers and without relocating placed items. We show that for any fixed load $\alpha < 1$, universal classes of $c \log n$-wise independent hash functions yield the same expected probe performance as fully random hash functions for **double** hashing, with an error of ϵ. That is, the expected number of probes to insert the αn-th item is $\frac{1}{1-\alpha} + \epsilon$ when $c \log n$-wise independent functions are used instead of idealized mathematically random functions. The positive constant c depends on α and ϵ, but not n. The error ϵ can be any fixed positive quantity. A consequence is that $O(\log\log m + \log^2 n)$ random bits suffice for these hash schemes. Moreover, subsequent work, which builds upon the theorems and lemmas of this paper, has reduced the error from ϵ to an optimal$O(\frac{1}{n})$ [17].

Our performance bound for double hashing readily applies to any generalization that exhibits approximate pairwise independence for the first $O(\log n)$ probes of any item, features statistically independent probe functions for any $O(\log n)$ items, and is robust in the sense that insertions must eventually succeed, provided the table is not full, and that probe locations cannot be revisited too often, during the insertion of an individual key. In these cases, the expected probe count to locate

the αn-th item is again bounded by $\frac{1}{1-\alpha}+\epsilon$. The performance bound for these generalizations is new even in the case of full randomness.

When combined with the highly independent fast hash functions of [16], these results give the first randomized classical closed hashing schemes featuring, for a word model of computation, a constant number of arithmetic operations per probe and nearly optimal probe performance.

These results are derived from a novel formulation that overestimates the expected probe count by underestimating the presence of local items already inserted into the hash table, and from a sharp analysis of the underlying stochastic structures formed by colliding items.

1.0 Introduction and background

Let $D = (x_1, x_2, \ldots, x_{\alpha n})$ be a sequence of αn distinct search keys, $\alpha < 1$, belonging to the universe $U = \{0, 1, \ldots, m-1\}$. We wish to hash D into a table $L[0..n-1]$. In closed hashing, which is also called open addressing, all data must be placed within the hash table, and pointers will not be allowed. In this model, each key $x \in U$ is mapped into a probe sequence $p(x,1), p(x,2), ... \in [0, n-1]$ (which ideally would be a permutation of $[0, n-1]$), and the generic insertion scheme is to place x in the first vacant table location in its probe sequence. The search procedure is to traverse the same sequence until the item is located, or an empty table slot is identified, in which case the item is known to be absent.

Uniform hashing is an idealized model where the probe sequence $p(x,*)$, for each key $x \in U$, is assumed to be a fully independent random function (or permutation). Traditional double hashing, which originates in the 1968 Ph.D. thesis of Guy de Balbine [10], defines $p(x,j) = f(x) - (j-1)d(x) \bmod n$, where the table size n is prime, $f(x)$ is assumed to return an arbitrarily selected integer in $[0..n-1]$, and $d(x)$ is an arbitrarily selected value in $[1,..n-1]$. The $2|U|$ random values $\{(d(x), f(x))\}_{x \in U}$ are assumed to be fully independent and uniformly distributed over their respective ranges.

In the most common versions of these hashing models, the probe sequences are used to place a key in its first vacant probe location, as opposed to some earlier position with the concomitant relocation of its former occupant. Relocation schemes, as originated by Brent (c.f. [4]), have been designed as a means to reduce the expected search time in applications where search operations are much more frequent than insertions. Such schemes, however, are not the subject of this study. Accordingly, we shall hereafter refer solely to models of closed hashing without rearrangement.

Contributions begun by Knuth, [10], Ullman [18], Ajtai, Komlós, and Szemerédi [2] culminate in a proof by Yao [20], who showed that in terms of retrieval cost, uniform hashing is optimal: no fixed set of hash functions can perform better than the random ones of uniform hashing. For double hashing, work by Guibas and Szemerédi [8] and subsequent results by Lueker and Molodowitch [11] culminate in a proof that for random functions f and d and any fixed load factor $\alpha < 1$, the expected number of probes to insert the $(\alpha n+1)$-st item is $\frac{1}{1-\alpha}+O(\frac{\log^{5/2} n}{\sqrt{n}})$, which is asymptotically equivalent to uniform hashing, and hence optimal.

An important consequence of these analyses is the certainty that only two random functions need to be defined to provide an optimal hashing scheme. Unfortunately, the question of what computable functions can be proven to behave like random hash functions has been open. The formal use of fully random functions on U leaves our understanding of computable hashing in an unsatisfactory state, since such a function has a program size (Kolmogorov complexity) of about $2|U|\log n$ bits, on average. We could use polynomials of degree $n-1$ to implement n-wise independent random functions (c.f. Definition 1) to reduce the spatial cost to $O(\log\log m+n\log n)$, while raising the time needed for evaluating the hash function to n, which is no better than the time needed to search an unordered list.

A more constructive perspective on the traditional analyses such as [11] is that they establish optimal performance bounds for hash schemes that use programmable functions, provided the measure of running time is taken to be the performance averaged over all possible input sequences. This is not the same as a randomized performance bound, where the expected running time for any fixed sequence of data is shown to be optimal. The problem of sorting makes this distinction especially clear. Suppose we wish to sort n integers in the range $[0, m-1]$. It is widely believed that no algorithm, when m is arbitrary, can run in linear time. Yet if integer division of $\log m$-bit words is taken to be a unit time primitive, then a Binsort of the data into the intervals $[i\frac{m}{n}, (i+1)\frac{m}{n}-1]$, for

$i = 0, 1, 2, \ldots$, followed by a sorting of each partition will run in linear time on the average, because the Binsort partitions the data into pieces exhibiting an average size of 1 and a variance slightly below 1. There are, of course, many data sets where the partitioning is useless.

Carter and Wegman [6] and [19], contributed to our understanding of limited randomness and randomized performance by introducing the notion of universal hash functions, and showing that these functions, when used for open hashing with separate chaining, result in an expected probe performance that is equivalent to the fully random formulation. In this model, $L[i]$ comprises a linked list of items hashing to the value i: external storage is used to hold colliding items and pointer information linking them together.

In particular, Carter and Wegman exhibited the universal classes of uniformly distributed h-wise independent hash functions, (which they called universal$_h$):

$$F_{h,n} = \{f \mid f(x) = (\sum_{0 \le j < h} a_j x^j \mod p) \mod n,\ a_j \in [0, p-1]\}, \tag{1}$$

where $p \ge m$ is prime. They showed that if, for any $D \subset U$, a hash function is randomly selected from $F_{h,n}$ (independent of D), then the sum of the expected j-th moments of the chain (i.e., list) lengths is essentially the same as that resulting from fully random functions, for $j \le h$. For separate chaining, the second moment determines the expected retrieval time, whence pairwise independence guarantees optimal expected performance.

Carter and Wegman also posed as an open question whether a comparable result could be achieved for any form of closed hashing, such as double hashing. We resolve the question affirmatively if, for a sufficiently large c, $c \log n$-wise independent hash functions are used.

For our purposes, however, the evaluation time of their universal hash functions is too high, but the results of [16] exhibit such families with constant evaluation time for a standard word model Random Access Machine. (The model in [16] follows the standard conventions of allowing indexed access to a size n array in constant time. A constant number of multiplications and integer divisions are deemed to require $O(1)$ time for keys in the universe U, but most of the (constant number of) operations are on $O(\log n)$-bit words. The requisite number of random bits still turns out to be $O(\log\log m + \log^2 n)$, although the $O(\log^2 n)$ dependence is increased by a constant factor, and auxiliary storage of n^ϵ $(\log n)$-bit words is provably necessary, for some $\epsilon < 1$.)

The attraction of a universal hashing formulation is two-fold. First, the notion provides a way to construct randomized algorithms for hashing, which eliminates any requirement that the data be "random," for suitable performance. Second, it allows a fixed – presumably computable – set of functions to be used for hashing, as opposed to an axiomatically specified "random" function.

This work differs considerably from the analyses of [8] and [11] in that both analyze the intersection patterns of arithmetic progressions, whereas this work has no notion of such a sequence. Byproducts of such a proof formulation include the following.

1) Automatic generalization of our performance results to arbitrary hashing schemes that satisfy (minor) requirements regarding adequate table coverage and that (more importantly) exhibit approximate pairwise independent probing, which can be formalized as follows.
   $\forall x\ \forall i \neq k\ \forall \ell \neq j:\quad Prob\{p(x,k) = \ell,\ p(x,i) = j\} \le \frac{1}{(n-O(1))^2}$.[†]
   Actually, the probe sequence can be defined by the equation $p(x,j) = h(f(x), d(x), j)$, for random f and d, and any **deterministic** function h as long as pairwise independence is assured for the first $O(\log n)$ probes, and reasonable coverage occurs for later probes. The coverage requirements, as quantified later, simply ensure that overall probe coverage is adequate to guarantee that the insertion of a key x will fail only when the table is full, and that within any x's probe sequence, locations will not be repeated enough to degrade the resulting performance.

2) A proof that $O(\log n)$-wise independent hash functions are random enough to preserve the expected performance of the hashing schemes in 1), up to a fixed error of ϵ.

The rest of this section is organized as follows. Subsection 1.1 explains why a two-level hashing scheme can enable the use of functions with a spatial complexity of only $O(\log\log m + \log^2 n)$ random bits. Subsection 1.2 formalizes a notion of limited independence with requirements that, in most respects, are slightly stronger than the definitions generally encountered in the literature (and in

[†] We use the Big-Oh notation in the following standard way: $f = g + O(h)$ means that $|f - g| = O(|h|)$. Consequently, there is no distinction between $f + O(g)$ and $f - O(g)$. Nevertheless, we shall, upon occasion, use minus signs to suggest that the worst case error is negative. Also, it is quite reasonable to write, say, $\frac{1}{1+O(h)} = 1 - O(h)$, for $h = o(1)$.

other respects slightly weaker) but are still readily achievable. It also presents constructions of two-level hash functions that exhibit the statistical randomness required by our analyses. Subsection 1.3 outlines the rest of the paper.

1.1 Reducing the domain

Although our proofs show that any set of sufficiently well behaved hash functions can be used for double hashing, it is worth noting that a universal class of linear congruential hash functions can be used to map the data D into a polynomial sized space such as $[0, n^4]$ in a collision-free manner, with high probability. Then the universal class $F_{h,n}$,(as defined in Section 1.0), can be restricted to have coefficients of size $O(n^4)$, as opposed to size $O(m)$, which might be much larger. Such mappings can be pieced together from techniques in [6], [12] and [7]. Accordingly, we first exploit the following variation of Lemma 2 from [7]:

Fact 1: Let $P_k = \{p \mid p \text{ is prime and } p \in (n^k \log m, (2+\beta)n^k \log m)\}$, for some small suitably fixed $\beta > 0$. Then

$$\forall x \neq y \in D: \ Prob_{p \in P_k}\{x = y \bmod p\} < n^{-k}.$$

Proof: [12],[7]. By the Prime Number Theorem, $|P_k| = \frac{(1+\beta)n^k \log m}{k \log n + \log \log m}(1 - o(1))$. The product of any $\gamma|P_k|$ primes in P_k is bounded below by $(n^k \log m)^{\gamma|P_k|} > (m)^{\gamma n^k}$, whence no more than $\gamma \leq 1/n^k$ of the elements of P_k can divide $|x - y|$. ▮

Fact 2: Let $F_0(p) = \{h \mid h(x) = (ax + b \bmod p) \bmod n^k, \ a \neq 0, b \in [0, p-1]\}$, where $p > n^k$ is prime. Then

$$\forall x \neq y \in [0, p-1]: \ Prob_{f \in F_0(p)}\{f(x) = f(y)\} \leq n^{-k}.$$

Proof: [6]. Given x and y, $x, y \in [0, p-1]$, $x \neq y$, the number of different $f \in F_0(p)$ where $f(x) = f(y)$, is precisely the number of 2×2 linear systems in a and b:

$$\begin{cases} ax + b = c + dn^k \bmod p, \\ ay + b = c + en^k \bmod p, \end{cases} \qquad \text{where } c, d, e \geq 0; \ c + dn^k < p; \ c < n^k; \ e \neq d; \ c + en^k < p.$$

Now $c + dn^k$ can have p different values. The remaining parameter e cannot be set to d because this would give $a = 0$. Thus there are at most $\lceil p/n^k - 1 \rceil$ different values available for e. Since there are exactly $p(p-1)$ different functions in F_0, and the number of f where $f(x) = f(y)$ is at most $p\lceil p/n^k - 1\rceil = p\lfloor p/n^k \rfloor \le p\frac{p-1}{n^k}$, the result follows. ▮

Combining Facts 1 and 2 shows that a hash function selected at random from $F_0^k = \cup_{p \in P_k} F_0(p)$ will, with probability exceeding $1 - 2\binom{\alpha n}{2} n^{-k}$, map D into $[0, n^k - 1]$ with no collisions at all among its $\binom{\alpha n}{2}$ pairs. We may take $k = 4$, so that the probability of a collision is below $1/n^2$, and assume the functions $F_{h,n}$ are defined for $p \approx n^4$.

Because of this preprocessing, the spatial complexity of our composite universal hash functions $F_{h,n} \circ F_0^4$ is $O(\log\log m + \log^2 n)$ bits, for $h = O(\log n)$.

1.2 Limited randomness

Since the randomness of our hash function family restricts the size of the small data sets where the hashing behavior is easy to analyze, it is convenient to formalize this family characteristic.

Carter and Wegman defined a family of hash functions F with domain U and range R to be strongly *universal*$_h$ if

$$\forall\ y_1, \ldots, y_h \in R, \forall \text{ distinct } x_1, \ldots, x_h \in U: \quad |\{f \in F : f(x_i) = y_i,\, i = 1, 2, \ldots, h\}| = \frac{|F|}{|R|^h},$$

so that the fraction of functions in F that achieve the desired mapping of the x_i's is the same as that for fully random functions. This definition combines the requirements of uniformity and h-wise independence. The specification is a little stronger than that used by Carter and Wegman for open hashing, and was introduced by them for application in cryptography [19]. They also gave an application of *almost universal*$_2$ functions where the function density $\frac{|F|}{R^h}$ is multiplied by a constant factor and used as an upper bound.

Our bounds for closed hashing are so dependent upon inclusion-exclusion that we need a very precise notion of *almost universal*$_h$, which separates the uniformity and independence requirements and which is formalized as follows.

Definition 1.

We say that a set of functions F with domain U and range R is an h-wise independent universal family of hash functions with *β-tolerance* if F exhibits

(h)-wise independence: $\forall\ y_1, \ldots, y_h \in R, \forall$ distinct $x_1, \ldots, x_h \in U$:

$$\frac{|\{f \in F : f(x_i) = y_i,\, i = 1, 2, \ldots, h\}|}{|F|} = \prod_i \frac{|\{f \in F : f(x_i) = y_i\}|}{|F|},$$

and near uniformity:

$$\forall\ y \in R, \forall\ x \in U : \quad (1 - \beta)\frac{|F|}{|R|} \le |\{f \in F : f(x) = y\}| \le (1 + \beta)\frac{|F|}{|R|}.$$

Thus the family of hash functions has a nearly uniform distribution, and the joint probability distribution on any subset comprising h or fewer points in U, exhibits the usual multiplicative independence. It is worth observing that the function classes $F_{h,n}$, from Section 1.0 are (h)-wise independent with $1/n^3$-*tolerance* for a universe of size n^4.

On the other hand, our premapping step for larger universes will not quite meet the multiplicative requirement of Definition 1 because F_0^4 will have too many hash functions that map a sequences of hash keys D into $[0, n^4]$ with collisions. If we ignore such unfortunate cases, and charge, say, an $O(n)$ cost per insertion for such instances, then our performance bounds will not change, and we may rely on the constructions of Section 2 to perform well enough in general. Accordingly, our final randomness characterization is as follows.

Definition 2.

We say that a family of hash functions F with domain U and range R is *effectively* (h)*-wise independent with* β*-tolerance* if for each $D \subset U$ with $|D| \le n$, $\exists \hat{F} \subset F$ where $\frac{(1+n\beta)|\hat{F}|}{|F|} > 1$ and $\hat{F}$ is $(h)_\beta$-wise independent with tolerance-β for domain D and range R.

We shall take the requirement of uniform distribution with β-tolerance to be understood, and simply refer to these schemes in terms of their limited independence. Section 1.1 gives a formal construction where for any fixed set $D \subset U$ of n input keys, all but $1/n^2$ of the F_0^4 map D into $[0, n^4]$ in a collision free way, and the subsequent hashing is fully (h)-wise independent with tolerance $\frac{1}{n^3}$. Evidently, this family is effectively (h)-wise independent with tolerance-$\frac{1}{n^3}$ according to the requirements of Definition 2.

In our formal models, a family of hash functions H comprises a finite set of functions. Given the data sequence D, a specific hash function is selected by randomly choosing a function from H

with each element equally likely to be selected. The statistical properties defined by Definitions 1 and 2, as well as the those which follow in Definition 4 are with respect to H.

1.3 A proof outline

Traditional analyses of hashing view the state of a hash table as a stochastic process that evolves over a duration of αn probabilistic insertions. Lueker and Molodowitch [11], for example, analyze double hashing in the fully random case with an elegant scheme that keeps the table distribution uniform by introducing moderately improbable randomizing insertions of fake items to correct the distribution at each insertion step. By vigilantly maintaining a fully random table distribution, they establish a simple proof that double hashing and uniform hashing exhibit comparable collision statistics in a fairly strong sense, and this intuition has turned out to be invaluable in this current work, which establishes an even closer statistical equivalence. Unfortunately, such an evolutionary approach seems to be inappropriate for instances where the randomness is limited, since all of the randomness would be used up after $\log n$ insertions. Instead, we are obliged to establish the bounds with a proof technique that can be extended from uniform hashing to double hashing with full independence to a comparable double hashing with limited independence. The hashing models are fully specified in Definition 4.

Let a fixed hashing model complete with (probabilistically selected) hash functions be specified, and consider a key $x \in D$. We may define its dependency set $dep(x, D)$ (Definitions 5,6, and 7) to comprise x and the recursively defined members of the dependency sets of the keys that occupy the table locations probed during the insertion of x.

Given a subsequence $S \subset D$, one may ask, what is the probability that the specific items in S are the precise and full cause for the number of probes needed to insert x? A necessary condition for S is that $dep(x, S) = S$ (c.f. Definition 8 as applied to $P(k, k)$ for $k = |S|$, and Section 2.1.1). The probability that the probe sequences for S have this behavior turns out to be, up to a factor of $(1 + O(\frac{|S|^3}{n}))$, the same in uniform hashing and our generalized double hashing schemes, as long as $3|S|$ does not exceed the amount of independence of our hash functions (Lemma 2).

Unfortunately, there can be many subsequences in D that, in the absence of other competing sequences, would satisfy the collision conditions for the set S. We may characterize, $dep(x,D)$ as that special S where $dep(x,S)=S$ and each $y\in S$ turns out to encounter no $z\in D-S$ residing in its probe locations when it is inserted as a member of the full sequence D: its probe locations must only contain elements from S (Lemma 3). The probability, that each y in S satisfies this latter criterion when all of D is hashed, is more difficult to estimate. We define the formal notion of a *multiplicative* vacancy estimator $q(t)$ (Definition 14) that gives an overestimate of the probability that a given location will be empty when x_t is hashed as the t-th element in D.

Then an explicit overestimate for the expected number of probes to insert $x_{\alpha n}$ can actually be calculated for uniform hashing and any *multiplicative* vacancy overestimator q (Theorem 1).

The most complicated calculation for double hashing is to estimate the probability that an arbitrarily specified sequence of probe locations $I\subset[1,2,\ldots,n-1]$ satisfies the claim $\forall j\le|T|:\ (L[I_j]$ is vacant prior to the insertion of $x_{\gamma_j})$ (Lemma 6 and Theorem 3). We define a quantifiable notion of weak vacancy estimator (Definition 15) where location ℓ is "vacant" at time t if no subsequence of h or fewer items in $x_1,x_2,\ldots,x_{t-1}$ hash into a local dependency set that embeds a key in $L[\ell]$. Then we formalize the notion of a witness set (Definition 16), which will comprise (a maximal) subset of $D-S$ that includes (among other keys) all subsets that will (or might) cause the vacancy condition to be false. The probability that our weakened vacancy criterion holds can then be estimated by a summation (equation (22)) over all subsets of $D-S$ of the probability that each subset is a witness set that does not contradict the vacancy statement, and this summation could, in principle, be summed (in equation (2)) to give an expression that overestimates the insertion cost for $x_{\alpha n}$. Rather than evaluate such a hopelessly complicated summation, we show that the sum is asymptotically the same for uniform hashing and double hashing with full independence (Lemma 6).

Inclusion-exclusion is used to extend the result to double hashing with limited independence. We also have to establish a bound that guarantees that the witness set has a size that is proportional to $\log n$, with overwhelming probability (Lemma 7).

Lastly, Theorem 3 shows that for uniform hashing, the explicit vacancy estimate given for q

satisfies the multiplicativity criterion used for our estimate in Theorem 1, which therefore holds, and provides an evaluation of our more complicated performance summation under all models.

2.0 Generic probe counts

Since our probe formulations are based on graphs that capture all essential collision behavior, a few preliminary definitions would seem to be appropriate.

2.1 Basic definitions

Definition 3.

- The hash keys $D = (x_1, x_2, \ldots, x_{\alpha n})$ comprise a sequence of αn distinct items, $\alpha < 1$, belonging to the universe $U = \{0, 1, \ldots, m-1\}$, and $p(x,j) : U \mapsto [0, n-1]$ denotes the j'th probe for key x.

- The ith element in a sequence S is denoted by S_i. For D, we also have $D_i = x_i$.

- The random variable giving the number of probes needed to insert x_i is defined to be $probe_i$. The randomness is due to the randomness in the hash functions as opposed to the data.

- *A rooted DAG* is a directed acyclic graph with only one root, (i.e. one vertex with indegree 0).

- Let $dgr(G)$ of a rooted DAG G be the outdegree of the root of G.

- Let x is *embedded* at location ℓ mean that as a consequence of hashing D into table L, $L[\ell] = x$. We extend this notion to include cases where only a subsequence $S \subset D$ of the data is hashed, in which case D should be replaced by S, and the embedding assignment should be understood to be possibly incorrect, when all of D is processed.

We will be analyzing how D is hashed into a table L of size n under three models: uniform hashing, a generalization of double hashing where random hash functions are used, and the same double hashing model where the hash functions are constructed from a family of (ψ)-wise independent family of universal hash functions.

Definition 4: The models UH, DH, and DH_ψ.

- In UH, the probe sequence $p(x,*)$ is an independent family of random variables that are uniformly distributed over $[0, n-1]$. Any collection of sequences $p(x_1,*), p(x_2,*), \ldots, p(x_n,*)$ are mutually independent, for distinct x_i.

- DH relaxes the requirement that each individual probe sequence be fully random.

  1. Each probe sequence $p(x,*)$ exhibits approximate pairwise independence:
  $$\forall x \ \forall i,j \ i \neq j \ \forall r,s \in [0,n-1] \ r \neq s : \quad Prob\{p(x,i) = r, \ p(x,j) = s\} = \frac{1}{(n-O(1))^2}.$$
  2. Furthermore, the random sequences $\{p(x,*)\}_{x \in D}$ are mutually independent. This condition need only hold for a subset of hash functions $\hat{F} \subset F$, where $\hat{F}$ depends on D, and $\frac{|\hat{F}|}{|F|} > 1 - \frac{O(1)}{n^2}$.
  3. In addition, we have the following robustness requirements.

     i) Extremely long probe sequences are quite rare: For a fixed c_0 that depends on α,
     $$\forall x : \sum_{t > c_0 n} Prob\{|\cup_{i=1}^{t} \{p(x,i)\}| < \alpha n + 1\} < \frac{O(1)}{n}.$$
     ii) Probe sequences are unlikely to reprobe locations too frequently.
     $$\forall x \ \forall j < k < h, r \in [0,n-1] : \ Prob\{p(x,j) = p(x,k), \ p(x,h) = r\} = \frac{O(1)}{n^2}.$$
     $$\forall x \ \forall h < i, j < k, \ (h,i) \neq (j,k) : \ Prob\{p(x,h) = p(x,i), \ p(x,j) = p(x,k)\} = \frac{O(1)}{n^2}.$$

- In DH_ψ, the statistical probe behavior of an individual probe sequence is subject to the same requirements as in DH for the first ψ probes, the global coverage requirement must still hold, and the joint distribution of initial probe sequences, for collections of ψ or fewer probes, is required to be statistically independent, for distinct items. More precisely,, we have the following.

  1. $$\forall x \ \forall i,j \leq \psi \ i \neq j \ \forall r,s \in [0,n-1] \ r \neq s : \quad Prob\{p(x,i) = r, \ p(x,j) = s\} = \frac{1}{(n-O(1))^2}.$$
  2. Knowing a limited number of the probe values for a small set of keys gives no information about the first few probes for another key. Formally, let Z be a set of keys $\zeta \in U$ with

associated probe count bounds j_ζ, where $\sum_{\zeta \in Z} j_\zeta \le \psi$. Let $\{\kappa_{\zeta,1}, \kappa_{\zeta,2}, \ldots, \kappa_{\zeta,j_\zeta}\}_{\zeta \in Z}$ be a multiset of arbitrary probe locations. Then

$$Prob\{ \bigwedge_{\zeta \in Z} \bigwedge_{j \le j_\zeta} p(\zeta, j) = \kappa_{\zeta,j} \} = \prod_{\zeta \in Z} Prob\{ \bigwedge_{j \le j_\zeta} p(\zeta, j) = \kappa_{\zeta,j} \}.$$

This condition need only hold for a subset of hash functions $\hat{F} \subset F$, where $\hat{F}$ depends on D, and $\frac{|\hat{F}|}{|F|} > 1 - \frac{O(1)}{n^2}$.

3. For some fixed c_0 that depends on α, $\forall x :\ \sum_{t > c_0 n} Prob\{| \cup_{i=1}^{t} \{p(x,i)\}| < \alpha n + 1\} < \frac{1}{n}$.
4. $\forall x\ \forall j < k < h \le \psi,\ r \in [0, n-1] :\ Prob\{p(x,j) = p(x,k),\ p(x,h) = r\} = \frac{O(1)}{n^2}$.
5. $\forall x\ \forall h < i \le \psi, j < k \le \psi,\ (h,i) \ne (j,k) :\ Prob\{p(x,h) = p(x,i),\ p(x,j) = p(x,k)\} = \frac{O(1)}{n^2}$.

The requirement that $r \ne s$ is explicitly included in 1 of DH to ensure that standard double hashing, which suffices to guarantee that $p(x,*)$ be a permutation, belongs within DH. Similarly, double hashing and all of DH are included in DH_ψ.

Our robustness requirements replace the (stronger) requirement that probe sequences be permutations. It is easily seen that some form of robustness is necessary to guarantee that hash functions do not fail. Consider the damage that would occur if the offset function $d(x)$, for standard double hashing, were allowed, for example, to be zero even with the tiny probability $1/n^3$: with probability $1/n^3$ the number of probes needed to insert x_i becomes ∞ and so does the expected probe count. Such degenerate functions must therefore be excluded from DH and DH_ψ.

Since we are using finite classes of hash functions, a single defective function can place an otherwise efficient algorithm outside DH or DH_ψ. On the other hand, we may include such classes in DH_ψ by modifying the hashing procedure when, say, an item's first $n^{1/3}$ probes (or $O(n)$) have failed to find a vacant location. A suitable strategy would be to switch to linear probing (where $p(x,j) = f(x) - j + 1 \bmod n$, for a random f), which would reduce the probability of failure to zero, and satisfy our global robustness requirement. We could even set $f \equiv 0$, in this case. Alternatively, one could select new random seeds and rehash the entire data set.

The independence ψ needed for these proofs will turn out to be $O(\log n)$. An immediate consequence of this work is that standard double hashing will achieve near optimal performance, if for example, the probe and offset functions f, d are chosen from an *effectively ψ-wise independent* family of hash functions with tolerance $\frac{1}{n^3}$. For example, an adequately independent family is given by $f \in F_{\psi,n} \circ g$, and $(d-1) \in F_{\psi,n-1} \circ g$, where g is a random function in F_0^4. Subject to the caveats needed to ensure the absence of failure, uniform hashing will achieve near optimal expected performance for the probe functions $p(x,j) = f(j+ng(x))$, for $g \in F_0^k$, and $f \in \bar{F}_{\psi,n}$, where $\bar{F}_{\psi,n} : [0, n^{k+1}] \mapsto [0, n-1]$, for, say $k=4$ as in Section 1.1. The function families $F_{\psi,n}$, $F_{\psi,n-1}$, and $\bar{F}_{\psi,n}$ could be the universal hash functions presented in Section 1.0, or the constant time functions of [16].

We have already seen that the presence of irregular hash functions, such as the small $1/n^2$ fraction in F_0^4 that have collisions on D are insignificant. We now drop all reference to them since they induce a probe cost of $O(1/n)$.

To analyze how the ordered data set D hashes into the table, we introduce a family of directed graphs to capture the structure of the collision events.

Definition 5: The dependency graph $G(D)$.

Given a sequence D of hash keys, we say that a hashing of D defines a directed *dependency graph* $G(D)$ as follows. The vertex set of $G(D)$ is D and the edge set is initially empty. Suppose that when inserting x into the hash table, x is placed in its k-th probe location l_k, (after probing $l_1, \ldots, l_{k-1}$). We add a directed edge from x to each of the items $x_{T_1}, \ldots, x_{T_{k-1}}$ residing in table locations $l_1, \ldots, l_{k-1}$. Each edge is labeled with its corresponding probe number, and each vertex $x_T \in D$ bears its label T, which is its position in the sequence D.

Notice that $G(D)$, despite extensive labeling, bears no information to indicate where nodes are embedded.

Definition 6: The dependency graph $G(x, D)$.

- The *dependency graph* of x in D, $G(x,D)$, is the restriction of $G(D)$ to the vertex set comprising x and all nodes reachable from x in $G(D)$. Its edges and vertices are both labeled.

- The *dependency set* of x, $dep(x,D)$, is defined as the vertex set of $G(x,D)$.

It is also convenient to refine these objects based upon intermediate events.

Definition 7: Partial dependency graphs $G_r(x, D)$.

- The *partial dependency* graphs of x_i are the $probe_i$ subgraphs of $G(x_i, D)$, in which x_i is restricted to prefixes of its probe sequence: $G_0(x_i, D) \subset G_1(x_i, D) \subset \cdots G_{probe_i - 1}(x_i, D) = G(x_i, D)$. $G_r(x_i, D)$ is composed of x_i, the edges corresponding to the first r probes of x_i and the restriction of $G(D)$ to the vertex set reachable from x_i by these r probes. We note that the graphs $G_r(x, D)$ and $G_{r+1}(x, D)$ might have the same vertex set and only differ in the outdegree of the root x in the graph.
- The vertex set of $G_r(x, D)$ is denoted by $dep_r(x, D)$.

The set of all partial dependency graphs of x is denoted by $\mathcal{G}^*(x, D)$. The vertices in each of these graphs have labels in $[1, |D|]$. We may relabel the vertices of any $G = (V, E), G \in \mathcal{G}^*(x, D)$, in the unique order preserving way to $1, 2, \ldots, |V|$. Let $\mathcal{G}(x, D)$ be the resulting set of relabeled graphs. Clearly $|\mathcal{G}(x, D)| = |\mathcal{G}^*(x, D)|$.

These definitions provide immediate formulations for the expected number of probes to insert x_i, as a function of its dependency graph.

We will count the expected number of partial dependency DAGs rooted at $x_{\alpha n}$, which means that root $x_{\alpha n}$ may not yet have found a vacant table slot for insertion. Thus the next probe, on behalf of $x_{\alpha n}$, will add another branch to the DAG, if the new slot turns out to be occupied. Let $x_{\alpha n}$ have r children in the DAG $G(x_{\alpha n}, D)$. Then it will have encountered $r + 1$ DAG s. (The first will have zero children since we do not require the root to be inserted when counting these structures.) Thus the number of such DAGs actually encountered by $x_{\alpha n}$ is precisely the number of probes needed to insert the key.

$$\begin{aligned} \mathrm{E}[probe_i] &= \sum_{k \geq 0} Prob\{dgr(G(x_i, D)) \geq k\} \\ &= \mathrm{E}[|\mathcal{G}(x_i, D)|]. \end{aligned}$$

To estimate the probability that a given labeled graph G with k vertices is in $\mathcal{G}(x_i, D)$, we can sample all subsequences S of $(x_1, x_2, \ldots, x_{i-1})$ with $k - 1$ vertices and map them as prescribed to the vertices of G. We may then evaluate the probability that the collision behavior of these vertices is exactly as prescribed, **and** that the chosen subsequence *is the right one* (i.e. the elements in S end up in the same locations regardless of whether only S or all of D is hashed). The probability of the latter event is clearly the more difficult to estimate, since it concerns all of D. Moreover, estimating the probability that a sequence is the right one involves estimating the probability that certain locations are not occupied at certain times, which is our original problem. There is, however, one important difference: we may overestimate the probability that a sequence is the right one by underestimating the probability that the locations in which a candidate subsequence is embedded are full. The resulting expected number of such S gives an overestimate for $\mathrm{E}[probe_i]$.

A key to determining a window size (of subsequences to examine) is to find a minimal sized h: $Prob\{|dep(x_{\alpha n}, D)| > h\} < 1/n^2$. Pursuant to this objective, we have the following.

Definition 8: The probabilities $P(k,j)$ and $\widetilde{P}(k,j)$.

- Let $P(k,j)$ be the probability that a partial dependency graph of x_j, (the jth item to be hashed), contains exactly k vertices: $P(k,j) = Prob\{|dep_r(x_j, D)| = k\}$, for some r.
- Let $\widetilde{P}(k,j)$ be the expected number of partial dependency graph of x_j that contain exactly k vertices: $\widetilde{P}(k,j) = \sum_r Prob\{|dep_r(x_j, D)| = k\}$.

The technical reason for defining the $\widetilde{P}(k,j)$ as an expected number as opposed to a probability is that a single dependency graph may have two partial dependency graphs with the same number of vertices, due to a collision between x_j and some node already within its dependency graph. Moreover, we now have the following formulation, which expresses $\mathrm{E}[probe_j]$ as a function of its partial dependency graphs.

Lemma 1. $\mathrm{E}[probe_j]$, the expected number of probes needed to insert the jth element x_j, equals $\sum_{0<k} \widetilde{P}(k,j)$.

Proof:

$$\sum_{0<k} \widetilde{P}(k,j) = \mathrm{E}[|\mathcal{G}(x_i, D)|]. \quad \blacksquare$$

Unfortunately, $\widetilde{P}(k,j)$ may be a little unruly in DH, because of the possibility of reprobing earlier probe locations. Accordingly, we account for such events as follows.

Definition 9.

Let $Err_0(j)$ be the expected number of probes from x_j to vertices already belonging to partial dependency sets of x_j: $Err_0(j) = \sum_{r=2}^{probe_j - 1} Prob\{dep_r(x_j, D) = dep_{r-1}(X_j, D)\}$.

Lemma 1 may now be restated as follows.

Corollary 1.

$$\mathrm{E}[probe_j] = Err_0(j) + \sum_{0<\ell} P(\ell, j). \quad \blacksquare$$

Remark 1.

Note that the probabilities $P(\ell, j)$ (and, in fact, all performance statistics) are defined with respect to a universal class of hash functions. Now, these probabilities are not, of course, exactly the same for all classes in DH_ψ or all classes in DH. However, since $UH \subset DH \subset DH_\psi$, Corollary 1 shows that DH_ψ would be guaranteed to provide optimal performance if $P(\ell, j)$ and $Err_0(j)$ were shown to be asymptotically the same for all families DH_ψ. In the following subsection, we establish that $P(k, k)$ is indeed essentially the same for all members of DH_ψ, when $\psi \geq 3k$.

2.1.1 The importance of $P(k, k)$

The value $P(k, k)$ is of special interest because the event $(|dep(x, D)| = k)$ corresponds to the existence of a subset $\delta \subset D$ of k items $x \in \delta$, which has the collision behavior $|dep(x, \delta)| = k$. Consequently, δ is the dependency set $dep(x, \widetilde{D})$ for data the subset $\widetilde{D} = \delta$. The event $dep(x, \delta) = \delta$, as the next lemma will show, is nearly independent of the hashing scheme and specific items being inserted. But before analyzing the probability distribution of dependency DAGs, we need a standard traversal procedure to extract unique spanning trees from each DAG.

Definition 10.

Given a DAG $G = (V, E)$, let its *spanning tree* be constructed according to the following process. Its vertices are scanned in order of decreasing index in D. When a vertex x is scanned, its children are immediately processed in order of decreasing probe count, so that the vertex in x's first probe location is processed after all of its siblings. The tree edges out of x will comprise the edges to x's previously unprocessed children.

Lemma 2. In DH_ψ, for $\psi \geq 3k$, $P(k,k)$ is within a factor of $(1+O(k^3/n))$ of the same value, for all k-tuples of distinct items in D and any family that satisfies the requirements of DH_ψ. More precisely, let G be a dependency tree of k vertices, and let $S=(S_1,S_2\ldots S_k)$ be a sequence of k distinct elements in U. Then for $k=O(n^{1/3})$,

1) The probability that the dependency graph $G(S)=G$ under DH_ψ, for $\psi \geq 3k$, is $\frac{1+O(k^2/n)}{n^{k-1}}$.

2) The probability that $G(S)$ is a rooted dependency DAG, that properly contains the structure G as its spanning tree with root S_k, under DH_ψ, for $\psi \geq 3k$, is bounded by $O(k^3/n^k)$.

Proof: Let $G(S)=(V,E)$, where $|V|=k$. Let the sequence $\widehat{S}$ be $(\widehat{S_1},\widehat{S_2},\ldots,\widehat{S_k})$, be the order the vertices are processed as genuine children in our spanning procedure, with the root placed first. Let $Tr=(V,E_{Tr})$ be the tree discovered by the search. We embed the vertices in the order of exploration, root first.

1) Suppose that $G(S)$ is a tree, whence $E_{Tr}=E$. If the root $\widehat{S_1}$ has h children then $\widehat{S_1}$ is embedded in its $h+1$st probe location, which is any one of n locations. The h children correspond to the h items $\widehat{S_1}$ encountered when it was inserted. The probability that these first $h+1$ probes are to distinct locations is between 1 and $1-O(h^2/n)$. Subsequent node embedding will have two constraints: a node with h children has its $(h+1)$-st probe location predetermined, and the first h probes must be to h distinct unembedded locations. Let R_j be the set of locations used for the tree node destinations that are specified prior to the placement specification for the children of node $\widehat{S_j}$, and let r_j be the location for $\widehat{S_j}$.

The probability that a node $\widehat{S_j}$ with h_j children hashes to meet these two constraints is:

$$Prob\left\{(p(\widehat{S_j},h_j+1)=r_j)\bigwedge_{1\leq i<\ell\leq h_j}(p(\widehat{S_j},i)\neq p(\widehat{S_j},\ell)),\ \bigwedge_{1\leq i\leq h_j}(p(\widehat{S_j},i)\notin R_j)\right\},$$

which is at most $Prob\{(p(\widehat{S_j},h_j+1)=r_j)\}\leq\frac{1+O(1/n)}{n}$, and at least

$$Prob\left\{(p(\widehat{S_j},h_j+1)=r_j)\right\}-\sum_{1\leq i<\ell\leq h_j}Prob\left\{(p(\widehat{S_j},h_j+1)=r_j),(p(\widehat{S_j},i)=p(\widehat{S_j},\ell))\right\}$$
$$-\sum_{\substack{1\leq i\leq h_j\\ r\in R_j}}Prob\left\{(p(\widehat{S_j},h_j+1)=r_j),(p(\widehat{S_j},i)=r)\right\},$$

which is bounded below by $\frac{1}{n+O(1)} - \frac{O(h_j^2)}{n^2} - \frac{kh_j}{(n-O(1))^2}$. We used our assumption of local robustness to derive the second term and pairwise independence to derive the third term. We appeal to the independence of individual probe sequences to multiply all k factors to get a value between $(1 - \frac{O(k^2)}{n-1})(\frac{1}{n})^{k-1}$ and $(\frac{1}{n})^{k-1}(1 + O(k/n))$, which proves 1).

2) If $G(S)$ is not a tree then $E \neq E_{Tr}$ and the nodes of Tr have different embeddings since collisions occurred. The tree construction is similar, but some nodes $x \in Tr$, will have gaps in their probe sequences $p(x, 1), p(x, 2), \ldots$ to their tree children, since edges to nodes that are already embedded or that have embedding locations already specified will be omitted. Now, the initial probe sequences for any k items are mutually independent, as long as the total number of probes is bounded by ψ. Consequently, the probability that V hashes into a DAG that yields E_{Tr} as its spanning tree is at most $\prod_{j=1}^{k} pr_j$, where pr_j *overestimates* the probability that the j-th vertex is hashed to have the correct probes to previously determined locations.

Let $\widehat{S_j}$ have h_j tree edges. To upper bound pr_j, we distinguish among three cases: $\widehat{S_j}$ has no non-tree edges, $\widehat{S_j}$ has fewer than $h_j + 2$ non-tree edges and at least one, and $\widehat{S_j}$ has at least $h_j + 2$ non-tree edges. Note that if no two locations can be probed twice in a probe sequence – as is the case in double hashing – then cases two and three combine into the case $\widehat{S_j}$ has at least one and at most $k - 1$ non-tree edges.

The first case is as the overestimate in 1), and contributes a probability of at most 1 to pr_1, and at most $\frac{1}{n}(1 + O(1/n))$ to pr_j, for $j > 1$.

In the second case, there are different DAG structures, depending on which probe count within $(h_j + 2, \ldots, 2h_j + 2)$ is the last and actually embeds $\widehat{S_j}$. Summing over all possible last probe counts, over the possible probe counts that correspond to the first non-tree edge, which is among the first $h_j + 1$ probes, and the set of possible destinations for this first non-tree edge, (which must be to a location already probed by $\widehat{S_j}$ or some other item in S), we get $\frac{O((h_j+1)(h_j+1)k)}{n^2}$ as an overestimate for the probability contributed to pr_j by case 2.

In the third case, there must be two consecutive non-tree edges among the first $2h_j + 2$

probes of $\widehat{S_j}$. These edges may go to previously embedded items or collide with earlier probes of $\widehat{S_j}$. To estimate this contribution to pr_j, we ignore the requirement that $\widehat{S_j}$ must be placed successfully and focus on the expected number of ways a first pair of such probes could occur, which is bounded by $(2h_j+1)\frac{O(k^2)}{n^2}$.

Combining like terms from the three cases into factors and multiplying gives

$$\prod pr_j = \prod_{1<j\leq k} (\frac{1}{n} + O(\frac{(h_j+1)k^2}{n^2})) = (\frac{1}{n})^{k-1}(1+O(k^3/n)),$$

and hence the probability that G results from the traversal of a non-tree DAG is at most $(\frac{1}{n})^{k-1}(1+O(k^3/n)) - (1-\frac{O(k^2)}{n-1})(\frac{1}{n})^{k-1} = O(k^3/n^k)$ ▮

Notice that we have used the pairwise independence of probes, the independence of probes for k different items, the local robustness requirements that restrict an item's probe sequence from excessive reprobing of previously tried locations, and have assumed that the insertion procedure did not fail. The total number of probes, which governs the independence ψ as defined in Definition 4 is less than $3k$. Even sharper bounds can be attained (more naturally) for true double hashing and for uniform hashing, but such results cannot improve our asymptotic efficiency results.

In view of Lemma 2, we need to examine the hash statistics associated with trees in greater detail. Accordingly, we have the following definitions.

Definition 11.

- Let $N(k)$ be the number of distinct ordered (dependency) trees that can occur with k vertices.
- Let $P_{tree}(k,j)$ be the probability that some partial dependency graph of x_j is a *tree* of k nodes.

Remark 2.

Lemma 2 shows that for any k element subset $\delta = (\delta_1, \ldots, \delta_k) \subset D$, where $k = O(n^{1/3})$:

(a) $P(k,k) = Prob\{dep(\delta_k, \delta) = \delta\} = n^{-k+1}N(k)(1+O(k^3/n))$

(b) $P_{tree}(k,k) = n^{-k+1}N(k)(1+O(k^2/n))$.

The next step is to formalize these remarks and to introduce $Err(k,j)$, a bound that will replace $Err_0(j)$, in the formula of Corollary 1, and include a truncation error that permits the $P(\ell,j)$ to be summed through the first k terms only.

Definition 12.

- Redefine $P(k,k)$ to be $n^{-k+1}N(k)(1+O(k^3/n))$.
- Let $P_{tree}(k,k) = n^{-k+1}N(k)(1+O(k^2/n))$.
- Let $Err_1(k,j)$ be the probability that the vertex count $|dep(x_j,D)| > k$.
- Let $RR(k,j)$ be the Boolean indicator function for the event
  ($|dep(x_j,D)| \leq k$ and some unsuccessful probe for x_j does not increase the size of its partial dependency set):
  $RR(k,j) = (|dep(x_j,D)| \leq k) \wedge (dep_r(x_j,D) = dep_{r+1}(x_j,D)$ for some $r \leq probe_j - 2)$.
  Let $Err_2(k,j) = 2\mathrm{E}[|dep(x_j,D)| \times RR(k,j)]$, so that we take a penalty of $2|dep(x_j,D|$ probes when x_j has a dependency set of size k or less, x_j has a directed edge to x_ℓ and the indegree of x_ℓ is greater than 1, in $G(x_j,D)$.
- Let $Err_3(k,j)$ be the probability that $|dep(x_j,D)| \leq k$ and x_j has at least $2|dep(x_j,D)|$ probes to $dep(x_j,D)$.
- Let $Err_4(j) = \sum_{t \geq c_0 n} Prob\{probe_j > t\}$, where c_0 is used in Definition 4 for DH and DH_ψ.
- Let $Err(k,j) = c_0 n Err_1(k,j) + Err_2(k,j) + c_0 n Err_3(k,j) + Err_4(j)$.

It is easy to see that $Err_4(j) = \sum_{t \geq c_0 n}(t+1-c_0 n) Prob\{prob_j = t\}$, so that this error is the expected number of probes beyond $c_0 n - 1$. The expected number of excess probes among the first $c_0 n$ are overcounted by the three other terms comprising $Err(k,j)$.

Corollary 2. For $\psi \geq 3k = O(n^{1/3})$,

$$\mathrm{E}[probe_j] \leq Err(k,j) + \sum_{1 \leq i < k} P(i,j).$$

Proof: In view of Lemma 1, we need only show that $Err_0(j) + \sum_{i>k} \widetilde{P}(i,j) \leq Err(k,j)$. But this follows from the definition of $Err(k,j)$. ∎

The following definitions will enable us to formulate $P(k,\alpha n)$ as a function of $P(k,k)$.

Definition 13.

Let I be a sequence of k distinct locations in our hash table. Let T be an increasing sequence of k indices, $T_k < \alpha n$, with corresponding items D_T in the ordered data set D.

- Define $loc_D(D_T)$ to be the sequence of table indices occupied by D_T when D is hashed into L. Let, for a data sequence D', $loc(D') = loc_{D'}(D')$, so that loc without a subscript, takes its argument to be the complete sequence being hashed.
- Let $M(T, I)$ be the event: all of D can be successfully hashed into L according to the following modified hashing process: for $j \notin T$, x_j is hashed according to its specified probe sequence; for $j = 1, 2, \ldots, |T|$, $x_{(T_j)}$ can be placed in the (formerly vacant) location $L[I_j]$ without concern for the probe sequence. If some $L[I_j]$ turns out to be already occupied at time T_j, then $M(T, I)$ does not occur.

  Thus $M(T, I)$ depends on I, T and $D - D_T$, but is independent of the values comprising D_T. Simply stated, $(loc_D(D_T) = I) = ((loc(D_T) = I) \wedge M(T, I))$. In models UH and DH the events $(loc(D_T) = I)$ and $M(T, I)$ are independent, although they depend on I.
- Let $q(T, I)$ denote the probability of $M(T, I)$. As noted earlier, we have not shown yet that the probabilities $q(T, I)$ for different families in DH_ψ are very close.
- Given any sequence δ, let $\delta||x$ denote the sequence δ with x appended at the end.

These definitions can now be put to use to find additional formulations for the expected number of probes.

Lemma 3.

In UH, for any fixed $I_0 \subset [0, n-1]$, with $|I_0| = k - 1$:

$$P(k, \alpha n) = P(k, k) \sum_{\substack{T \subset [1, \alpha n - 1] \\ |T| = k-1}} q(T, I_0). \tag{2}$$

In DH:

$$P(k, \alpha n) \leq P(k, k) \sum_{\substack{T \subset [1, \alpha n - 1] \\ |T| = k-1}} \max_{\substack{I \subset [0, n-1] \\ |I| = k-1}} q(T, I). \tag{3}$$

In DH_ψ, for $3k \le \psi$:

$$P(k,\alpha n) \le P(k,k) \sum_{\substack{T\subset[1,\alpha n-1]\\ |T|=k-1}} \max_{\substack{I\subset[0,n-1]\\ |I|=k-1}} Prob\{M(T,I) \mid (dep(x_{\alpha n}, D_{T||\alpha n}) = D_{T||\alpha n}) \wedge (loc(D_T)=I)\}, \tag{4}$$

$$\le P(k,k) \sum_{\substack{T\subset[1,\alpha n-1]\\ |T|=k-1}} \max_{\substack{I\subset[0,n-1]\\ |I|=k-1}} q_{\psi-3k}(T,I), \tag{5}$$

where the subscript in the expression $q_{\psi-3k}$ in (5) is intended to restrict numerical computation of q to inclusion-exclusion calculations that use no more than $(\psi-3k)$-wise independence.

Proof: In all three models:

$$P(k,\alpha n) = \sum_{\substack{|I|=k-1\\ |T|=k-1}} Prob\{(dep(x_{\alpha n}, D_{T||\alpha n}) = D_{T||\alpha n}) \wedge (loc(D_T)=I) \wedge M(T,I)\}$$

$$\le \sum_{|T|=k-1} \Big(\Big(\sum_{|I|=k-1} Prob\{(dep(x_{\alpha n}, D_{T||\alpha n}) = D_{T||\alpha n}) \wedge (loc(D_T)=I)\}\Big) \times \max_{\substack{I\subset[0,n-1]\\ |I|=k-1}} Prob\{M(T,I) \mid (dep(x_{\alpha n}, D_{T||\alpha n}) = D_{T||\alpha n}), (loc(D_T)=I)\}\Big).$$

Now $\sum_{|I|=k-1} Prob\{(dep(x_{\alpha n}, D_{T||\alpha n}) = D_{T||\alpha n}) \wedge (loc(D_T)=I)\} = Prob\{(dep(x_{\alpha n}, D_{T||\alpha n}) = D_{T||\alpha n})\}$. We know (Lemma 2 and Remark 2) that in all three models, $Prob\{dep(x_{\alpha n}, D_{T||\alpha n}) = D_{T||\alpha n}\} = P(k,k)$, for all $D_{T||\alpha n}$, and the same (asymptotic) equality holds if we add the restriction that the dependency graph has no more than $3k$ probes. Inequality (4) now follows.

In UH, the event $M(T,I)$ is independent of $(dep(x_{\alpha n}, D_{T||\alpha n}) = D_{T||\alpha n}) \wedge (loc(D_T)=I)$, and is uniformly distributed over all sets I that comprise $k-1$ elements. Hence (2) follows with equality.

In DH, $M(T,I)$ depends on locations I, but for any fixed I, is also independent of $\{(dep(x_{\alpha n}, D_T^{\alpha n}) = D_T^{\alpha n}) \wedge (loc(D_T)=I)\}$, since hash values on D_T do not disclose any information about hash values on $D - D_T$. Thus (3) follows.

In DH_ψ, large events are not necessarily independent, but our estimates of the conditional $M(T,I)$ will be based on windows of $\psi - 3|D_T|$ probe events for keys in $D - D_T$, conditioned on information about the hash function's behavior on D_T. Since these events are independent of the conditioning, the numerical estimate in (5) now follows. ▮

Further analysis of $P(k,k)$ will show that $P(k,\alpha n)$ is negligible for $k > C\log n$, for a suitable constant C. Similarly, $Err(k)$, will turn out to be negligible. As a consequence, the behavior of the hash function can be analyzed by determining $P(k,k)$ for the very light loads $k = O(\log n)$ and estimating $q(T,I)$ based on small samples of points in $D - D_T$, for small $|T| = O(\log n)$.

3. Good vacancy estimators and their generic performance equation

Suppose that T is a sequence of insertion times for a collection of keys that locally hash into a dependency graph. This dependency graph is important if its apparent hash locations are really empty at respective times T. We need a function $q(T)$ that overestimates the probability, in DH, that these $|T|$ locations are empty at times T.

Definition 14: Multiplicative vacancy overestimators.

- We call a Boolean function $M'(T,I)$ a vacancy overestimator if, for any sequence of table locations I and key indices T, the event $M(T,I)$, (that locations I are empty at respective times T), implies the event $M'(T,I)$.
- We call a function $q(t)$ a *multiplicative vacancy overestimator* for an event $M'(T,I)$ if q is decreasing and for any fixed $\alpha < 1$, some bound k, and for all $D : |D| = \alpha n$, $T \subset [1, \alpha n] : |T| \le k$, $I \subset [1,n] : |I| = |T|$, the following holds:

$$\max_I Prob\{M'(T,I) \mid [dep(x_{\alpha n}) = D_T] \wedge [loc(D_T) = I)]\} \le \left(1 + \frac{O(|T|^2)}{n}\right) \prod_{i=1}^{|I|} q(T_i).$$

We could have defined weaker multiplicative overestimators that have a correction factor of $(1 + \frac{O(|T|^\rho}{n})$, for some fixed $\rho > 2$ instead of $\rho = 2$, and our asymptotic results, it turns out, would be unchanged; this extra freedom, however, appears to be unnecessary.

Of course, any multiplicative overestimator for an event M' that overestimates M is also a multiplicative overestimator for M. We shall eventually take the bound $|T| \le k$ to be proportional to $\log n$, but shall adopt the expedient of leaving its value unspecified as long as possible. In any case, $(1+O(1/n))q(t)$ is an overestimate of the probability that a given table location is empty after $t-1$ items have been entered, (since we may choose $T = \{t\}$). Moreover, such q's do exist: $q(t) = 1$ is certainly a (very uninteresting) multiplicative overestimator. Given a multiplicative estimator q to overestimate the probability that a given dependency graph hashes into empty locations at the times specified by T, Lemma 1, Lemma 2 and Corollary 2 show that the expected probe count for x_j can be overestimated by a computation that is virtually identical in UH, DH, and DH_ψ, provided that for a suitable $k = O(\log n)$, $Err(k,j) = \frac{O(1)}{n}$, so that the computation can be restricted to dependency trees of size k or less. For DH_ψ, the independence ψ will be required to exceed $3k + s$, where the s-wise (conditional) independence of the probe sequences must guarantee that the behavior of q is as stated. Until these values are quantified, we shall expose the implicit dependencies by writing q_s and $s_{|I|}$ when appropriate.

The presentation of a suitably multiplicative overestimator q is technical, and will be deferred even further. Meanwhile, the reader may prefer to view the following development as if it were for UH, although the conclusions will apply to DH and DH_ψ as well.

Corollary 3. Let q_s be a multiplicative vacancy overestimator for UH, DH, or DH_ψ, where $s_k + 3k \le \psi$, and put $\frac{1}{n}\sum_{0<b<a} q_s(b) = Q_s(a)$. Then for $k = O(n^{1/3})$,

$$P_{tree}(k, \alpha n) \le (1 + \frac{O(k^2)}{n}) \sum_{\substack{|T|=k-1 \\ T\subset[1,\alpha n-1]}} P_{tree}(k,k) \prod_{i=1}^{k-1} q_s(T_i) \tag{6}$$

$$\le (1 + \frac{O(k^2)}{n}) P_{tree}(k,k) \frac{n^{k-1} Q_s(\alpha n)^{k-1}}{(k-1)!}. \tag{7}$$

Proof: Lemmas 2 and 3 show that

$$P_{tree}(k, \alpha n) \le (1 + \frac{O(k^2)}{n}) P_{tree}(k,k) \sum_{\substack{T\subset[1,\alpha n-1] \\ |T|=k-1}} max_I q_s(T, I).$$

Inequality (6) follows from Definition 14, which requires that $(1 + O(|T|^2)/n)\prod_i q_s(T_i)$ overestimate $q_s(T)$. Similarly, (7) is an immediate consequence, since $\frac{1}{(k-1)!}\left(\sum_{t=1}^{\alpha n-1} q_s(t)\right)^{k-1}$ includes, for all subsequences $T \subset [1, \alpha n]$ with $(|T| = k - 1)$, each of the products $\prod_{i=1}^{k-1} q_s(T_i)$, exactly once. ▮

Corollary 3 provides a way to estimate $P_{tree}(k, \alpha n)$ and hence E$[probe_{\alpha n}]$ from a vacancy estimator $q_s(t)$.

Theorem 1. Let q_s be a multiplicative vacancy estimator, for UH, DH, or DH_ψ, where $s_k + 3k \le \psi$

and $k = O(n^{1/3})$. Let $\frac{1}{n}\sum_{0<b<a} q_s(b) = Q_s(a)$. Then

$$\mathrm{E}[probe_{\alpha n}] < \frac{1}{\sqrt{1-2Q_s(\alpha n)}} + O(\frac{1}{n}) + Err(k, \alpha n).$$

Proof: By Lemma 2,

$$P_{tree}(k,k) = N(k)(\frac{1}{n})^{k-1}(1+O(k^2/n)), \tag{8}$$

where $N(k)$ counts the number of different dependency trees with the k vertices $x_1, x_2, \ldots, x_k$. To determine $N(k)$, we observe that the root of the dependency trees is fixed at x_k, the vertex with highest index. Both the rightmost subtree of x_k, as well as the tree consisting of root x_k plus the remaining vertices comprising its other subtrees, if any, constitute partial dependency trees. If the rightmost subtree contains j vertices, its elements can be chosen in $\binom{k-1}{j}$ ways. This results in the following recurrence equation for N:

$$N(1) = 1$$
$$N(k) = \sum_{1\le j\le k-1} \binom{k-1}{j} N(j)N(k-j),$$

which upon setting $j' = k - j$ gives:

$$= \sum_{1\le j'\le k-1} \binom{k-1}{j'-1} N(j')N(k-j').$$

Averaging these two formulations, and applying the equality $\binom{k-1}{j-1} + \binom{k-1}{j} = \binom{k}{j}$ gives:

$$N(1) = 1,$$
$$N(k) = \sum_{1\le j\le k-1} \frac{1}{2}\binom{k}{j} N(j)N(k-j), \quad k > 1. \tag{9}$$

Let

$$g(x) = \sum_{0<k} \frac{N(k)}{k!} x^k. \tag{10}$$

Then multiplying (9) by x^k, summing, and applying (10) gives:

$$g(x) = x + \sum_{k>1}\sum_{1\le j\le k-1} \frac{1}{2}\frac{N(j)}{j!}\frac{N(k-j)}{(k-j)!}x^k = g^2(x)/2 + x,$$

and hence

$$g(x) = 1 - \sqrt{1-2x} = 1 - \sum_{k\ge 0} \binom{1/2}{k}(-2x)^k. \tag{11}$$

Equating coefficients of x^k in (10) and (11) gives:

$$N(k) = k!\binom{1/2}{k}2^k(-1)^{k-1} = (k-1)!\binom{-1/2}{k-1}(-2)^{k-1}.$$

Substituting for $N(k)$ in (8), and using Corollary 3 to define $P(k,\alpha n)$ in terms of $P_{tree}(k,k)$ and $Q_s(\alpha n)$ gives:

$$P(k,\alpha n) \le \binom{-1/2}{k-1}\left(1+\frac{O(k^3)}{n}\right)\left(\frac{-2nQ_s(\alpha n)}{n-O(1)}\right)^{k-1} = \binom{-1/2}{k-1}\left(1+\frac{O(k^3)}{n}\right)\left(-2Q_s(\alpha n)\right)^{k-1}. \quad (12)$$

Summing the $P(i,\alpha n)$, as prescribed by Corollary 2 gives:

$$\begin{aligned} \mathrm{E}[probe_{\alpha n}] &< \sum_{i=1}^{k}(1+\frac{O(i^3)}{n})\binom{-1/2}{i-1}\left(-2Q_s(\alpha n)\right)^{i-1} + Err(k,\alpha n) \\ &< \frac{1}{\sqrt{1-2Q_s(\alpha n)}} + \sum_i(\frac{O(i^3)}{n})\binom{-1/2}{i-1}\left(-2Q_s(\alpha n)\right)^{i-1} + Err(k,\alpha n). \end{aligned} \quad (13)$$

The error attributable to non-tree DAGs is seen to be bounded by:

$$\sum_{i>0}\frac{O(i^3)}{n}\binom{-1/2}{i-1}\left(-2Q_s(\alpha n)\right)^{i-1} \le \sum_{i>3}\frac{O(1)}{n}\binom{-7/2}{i-4}\left(-2Q_s(\alpha n)\right)^{i-1} = O(\frac{(Q_s(\alpha n))^3}{n(1-2Q_s(\alpha n))^{7/2}}). \quad (14)$$

We conclude that for fixed $\alpha < 1$, and $2Q_s(\alpha n)$ bounded by some fixed value less than 1,

$$\mathrm{E}[probe_{\alpha n}] < \frac{1}{\sqrt{1-2Q_s(\alpha n)}} + O(\frac{1}{n}) + Err(k,\alpha n). \quad \blacksquare$$

It is worth remarking that the results and computations are monotone in q_s; any error in its estimation carries through in Q_s.

It is also reassuring to observe that Theorem 1 gives the correct performance bound for UH. In UH, the probability that a location is vacant at time αn is $1-\alpha+\frac{1}{n}$, and $Prob(M(T,I)) \le \prod_{i=1}^{|T|}(1-\frac{T_i-i}{n})$, whence $q(t) = 1-\frac{t-|I|}{n} = (1-\frac{t}{n})(1+\frac{O(|I|)}{n})$ is a multiplicative vacancy overestimator. In this case, $Q(\alpha n+1) = \frac{1}{n}\sum_1^{\alpha n}(1-\frac{j-|I|}{n}) = (\alpha-\alpha^2/2)(1+\frac{O(|I|)}{n})$.

Theorem 1 says that for UH,

$$\begin{aligned} \mathrm{E}[probe_{\alpha n}] &< \frac{1}{\sqrt{1-2(\alpha-\alpha^2/2)(1+O(|I|)/n)}} + O(\frac{1}{n}) + Err(|I|,n) \\ &< \frac{1}{\sqrt{1-2(\alpha-\alpha^2/2)}} + O(\frac{|I|}{n}) + Err(|I|,n) \\ &< \frac{1}{1-\alpha} + O(\frac{|I|}{n}) + Err(|I|,\alpha n). \end{aligned}$$

In fact, we ought to observe that

$$\mathrm{E}[probe_{\alpha n}] < \frac{1}{1-\alpha} + O(\frac{1}{n}) + Err(k, \alpha n).$$

This sharper bound follows by noting that $Q(t) = (\frac{t}{n} - \frac{1}{2}(\frac{t}{n})^2)(1 + \frac{O(j)}{n})$, for a partial dependency set of size j. Substituting this formulation into (12) gives a sum of error terms in (13) that are of the same form as (14).

We now use Theorem 1 to bound $Err(k, \alpha n)$.

Corollary 4. Let q be a multiplicative vacancy estimator, in DH and DH_ψ, for $s_k + 3k \le \psi$. Set $2Q(\alpha n) = \frac{2}{n}\sum_{0<j<\alpha n} q(j)$, and suppose that $2Q(\alpha n) < \beta < 1$. Then $Err(k, \alpha n) = O(\frac{n^2\beta^k}{\sqrt{k}(1-\beta)})$, and hence for $k > 3\frac{\log n}{\log(1/\beta)}$,

$$\mathrm{E}[probe_{\alpha n}] < \frac{1}{\sqrt{1 - 2Q(\alpha n)}} + O(\frac{1}{n}).$$

Proof:

1) Recall that $Err_1(k, j)$ is the probability that the vertex count $|dep(x_j, D)| > k$. We use $c_0 n$ as an overestimate of the first $c_0 n$ or fewer probes that occur, in this case, and show that $c_0 n Err_1(k, \alpha n) = \frac{O(1)}{n}$ for suitable k. So suppose that $|dep(x_{\alpha n}, D)| > k$. Then some $x \in D$ has a partial dependency set of size $\hat{k}$, where $k < \hat{k} \le 2k$. Indeed, let x_t be the first key in D to have a dependency set of size $k+1$ or more. Then each child of x_t in $G(x_t, D)$ can have a dependency set of size k or less. Sequencing over $G_1(x_t, D), G_2(x_t, D), \ldots$ gives a family of dependency graphs with vertex counts growing by steps of k or less, and eventually exceeding k. Hence one of these counts must be within $[k+1, 2k]$. It follows that $Err_1(k, \alpha n) < \sum_{i \le \alpha n}\sum_{k<j\le 2k} P(j, i) < \alpha n \sum_{k<j\le 2k} P(j, \alpha n)$.

The proof of Theorem 1 (inequality (12)) allows us to bound this sum as follows.

$$\begin{aligned} c_0 n Err_1(k,\alpha n) &< \alpha c_0 n^2 \sum_{j=k+1}^{2k} (1+O(j^3/n))\binom{-1/2}{j-1}(-2Q(\alpha n))^{j-1} \\ &< O(n^2)\sum_{j\ge k}\beta^j\prod_{i=1}^{j}(1-\frac{1}{2i}) \\ &< O(n^2)\sum_{j\ge k}\beta^j\prod_{i=1}^{j}e^{-1/2i} \\ &< O(n^2)\sum_{j\ge k}\beta^j e^{-(\log j)/2} \\ &< O(n^2)\sum_{j\ge k}\frac{1}{\sqrt{j}}\beta^j \\ &< O(n^2\frac{\beta^k}{\sqrt{k}(1-\beta)}). \end{aligned} \tag{15}$$

Taking $k = 3\frac{\log n}{\log(1/\beta)}$ gives an additive error of $\frac{O(1)}{n}$.

2) Recall that $Err_2(\alpha n, k)$ equals $2|dep(x_{\alpha n}, D)|$ in the case that $|dep(x_j, D)| \le k$ and x_j has some unsuccessful probe that does not increase its partial dependency set. Let $|dep(x_{\alpha n}, D)| = j$. There are at most j probes of $x_{\alpha n}$ that could be the first to revisit a dependency set. There are at most $j-1$ vertices that could be the probe's destination. Let the destination node be z. Key z can be reached by a direct probe edge from $x_{\alpha n}$, and by some earlier path from $x_{\alpha n}$ that comprises one or more edges. Let the node probed by $x_{\alpha n}$ along this path be w. If $w = z$, we have a constraint that two specific probes of $x_{\alpha n}$ are the same, which occurs with probability $\frac{1}{n-O(1)}$ for each of the $\binom{j}{2}$ or fewer possibilities. Otherwise, we compute the probability that the DAG structure hashed as specified by a traversal that begins with w, reaches all of its descendents, and then continues from $x_{\alpha n}$. The embedding of w is unconstrained, but $x_{\alpha n}$ will be constrained at two probe locations.

The expected number of ways these events can occur, in this case, is $N(j)\binom{j}{2}\frac{1}{n}n^{-j+1}(1+O(\frac{j^3}{n}))$, and hence $Err_2(\alpha n, k) \le \sum_{j=1}^{k} N(j)\frac{2j^3}{n}n^{-j+1}(1+O(\frac{j^3}{n}))$.

3) Recall that $Err_3(k, \alpha n)$ is the probability that $|dep(x_{\alpha n}, D)| = j \le k$ and $x_{\alpha n}$ has at least $2j$ probes to $dep(x_{\alpha n}, D)$. The probability that the dependency set $G(x_{\alpha n}, D)$ occurs as stated with

j nodes is bounded by $(1+O(j^3/n))N(j)P(j,j)2j^3/n^{j+1}$, since $x_{\alpha n}$ must have two consecutive probes among the first $2j$ that visit locations previously probed by $x_{\alpha n}$. We take $c_0 n$ as an estimate for the number of probes in this case.

4) Recall that $Err_4(\alpha n)$ is the expected number of probes of length $c_0 n$ or more needed to insert $x_{\alpha n}$. Then the expected number of probes contributed in this case is $c_0 n Prob\{$at least $c_0 n$ probes occur$\} + \sum_{t>c_0 n} Prob\{$at least t probes occur$\}$. The first term is already counted by $c_0 n Err_3 + c_0 n Err_1$. The second term is bounded by $\frac{O(1)}{n}$, according to our robustness requirement that long probe sequences be rare.

5) Recall that $Err(k,j) = c_0 n Err_1(k,j) + Err_2(k,j) + c_0 n Err_3(k,j) + Err_4(j)$. Taking $k = 3\frac{\log n}{\log(1/\beta)}$ gives an additive error of $\frac{O(1)}{n}$ for $c_0 n Err_1$. $Err_4 = O(1/n)$ by the global robustness requirement. As for $Err_2 + c_0 n Err_3$, summing these error terms over the range of dependency set sizes j gives a formulation that is equivalent to (14), and hence $O(\frac{1}{n})$ in size. ▮

We are now ready to identify our vacancy estimator.

Definition 15: The vacancy criterion $M^{(h)}(T,I)$ and its probability $q^{(h)}(T,I)$.

- Let $M^{(h)}(T,I)$, be the vacancy criterion: for $j = 1,2,\ldots,|I|$, no tuple $S \subset \{x_1,\ldots,x_{T_j-1}\} - D_T$ of size $|S| \le h$ hashes into a dependency **tree** G rooted at location I_j.
- Let $q^{(h)}(T,I)$ denote the probability that the vacancy criterion $M^{(h)}(T,I)$ holds.

Thus, $M^{(h)}(T,I)$ is a vacancy criterion with limited backtracking. The criterion deems a location ℓ to be occupied by time t if some witness subsequence S – comprising h or fewer items among the first $t-1$ elements – hashes locally into a dependency *tree* rooted at ℓ. Otherwise ℓ is deemed vacant. It should be noted that a witness sequence S may not represent the dependency graph actually rooted at ℓ. Moreover, it turns out that we will not actually determine $q^{(h)}(T,I)$; instead, we will estimate its value with moderate accuracy. For our calculations, the vacancy estimator will be virtually unaffected by the assumptions of limited independence, as well as the specific hashing model, but will be strong enough to give good hashing bounds when used in Theorem 1. We also note that $M^{(h)}(T,I)$ excludes small dependency sets that hash into a location I if the structure is not a tree or if the structure uses more than one probe to locations in I. These overestimates of the vacancy will turn out to be asymptotically negligible.

Lemma 4. For any fixed $\alpha < 1$ and fixed h, in uniform hashing,

$$q^{(h)}_{UH}(\alpha n, i) \le 1 - \alpha + \frac{1}{\sqrt{h+1}} \frac{(2\alpha - \alpha^2)^{h+1}}{(1-\alpha)^2} + \frac{O(1)}{n}.$$

Proof: If the event $M^{(h)}(T_j, I_j)$ occurs, then either I_j is vacant at time T_j or the true dependency graph rooted at I_j has at least $h+1$ vertices. The probability of the event $M^{(h)}(T_j, I_j)$ is therefore bounded by the probability that I_j is empty at time T_j plus the probability that the **true** dependency graph rooted at location I_j, at time T_j, is a DAG with $h+1$ or more vertices.

Such a dependency graph differs from the dependency graphs we have analyzed so far in just two respects. First, the root is required to be embedded at the fixed location I_j. Second, the root could be any key in $(x_1, x_2, \ldots, x_{T_j - 1})$.

Let $P_{loc-i}(k, T_j)$ be the probability that the true dependency graph rooted at location I_j by time T_j is a DAG with k nodes. It is easy to see that the computation for $P_{loc-i}(k, T_j)$ is very similar to that for $P(k, T_j)$. The reasoning of Lemmas 2 and 3 gives the following.

$$\begin{aligned} P_{loc-i}(k, \alpha n) &\le (1 + O(k^3)/n) \sum_{\substack{|T|=k \\ T \subset [1, \alpha n - 1]}} N(k) n^{-k} \prod_{i=1}^{k} q(T_i) \\ &\le (1 + O(k^3)/n) N(k) \frac{Q(\alpha n)^k}{k!} \\ &= -\binom{1/2}{k} \left(1 + \frac{O(k^3)}{n}\right) \left(-2Q(\alpha n)\right)^k. \end{aligned} \tag{16}$$

We have seen that in the uniform hashing, of UH, $2Q(\alpha n) \le 2\alpha - \alpha^2$, since the sum for Q includes fewer than αn terms. Hence for suitable K,

$$\begin{aligned} Prob_{UH}\{M^{(h)}(\alpha n, I_{\alpha n})\} &\le (1-\alpha) - \sum_{k=h+1}^{K} \binom{1/2}{k} \left(1 + \frac{O(k^3)}{n}\right) \left(-2Q(\alpha n)\right)^k + \frac{O(h^3)}{n} + Err_1(K, \alpha n) \\ &\le (1-\alpha) + \frac{(2\alpha - \alpha^2)^{h+1}}{\sqrt{h+1}(1-\alpha)^2} + \frac{O(1)}{n}, \text{ by the estimate in (15).} \quad \blacksquare \end{aligned}$$

It is worth remarking that our vacancy estimator actually convergences at a much faster rate (as a function of h) than the estimate given by Lemma 4. The Lemma used a bound for the probability that the true dependency graph rooted at a specific location has h or fewer vertices, as opposed to the probability that some witness DAG with h or fewer vertices can certify that the location is already occupied. When the case $h = 1$, for example, it is easy to verify that the probability that a location will not have been hit by a first probe of αn items is $1 - e^{-\alpha n} + \frac{O(1)}{n}$, which is already much smaller than the corresponding bound predicted by Lemma 4.

Assume for the moment that, as we will soon prove, $q^{(h)}(t)$ is a multiplicative vacancy overestimator that satisfies $q^{(h)}(t) \leq (1 - \frac{t}{n}) + \frac{(2\frac{t}{n} - (\frac{t}{n})^2)^{h+1}}{\sqrt{h+1}(1-\frac{t}{n})^2}$. Then this estimator can be used with simple error estimates from Taylor's Series to establish the following Corollary.

Theorem 2. Suppose that $q_s^{(h)}(t) = (1 - \frac{t}{n}) + \frac{(2\frac{t}{n} - (\frac{t}{n})^2)^{h+1}}{\sqrt{h+1}(1-\frac{t}{n})^2} + \frac{O(1)}{n}$ is a multiplicative overestimator in DH_ψ and hence $Err(k, \alpha n) = O(1)/n$, for large enough $\psi = O(\log n)$. Let $2Q_s(\alpha n) = \frac{1}{n}\sum_{t \leq \alpha n} q_s^{(h)}(t) \leq 2\alpha - \alpha^2 + 2\frac{(2\alpha-\alpha^2)^{h+1}}{\sqrt{h+1}(1-\alpha)}$, and choose h so that $2\alpha - \alpha^2 + 2\frac{(2\alpha-\alpha^2)^{h+1}}{\sqrt{h+1}(1-\alpha)}$ is closer to $2\alpha - \alpha^2$ than 1. Then

$$\mathrm{E}[probe_{\alpha n}] < \frac{1}{\sqrt{1 - 2Q_s(\alpha n)}} + \frac{O(1)}{n} < \frac{1}{1-\alpha} + O(\frac{(2\alpha - \alpha^2)^{h+1}}{\sqrt{h+1}(1-\alpha)^{5/2}}) + \frac{O(1)}{n} \quad \blacksquare$$

This theorem is actually our main result. The remainder of this paper is solely aimed at showing that the probability of $M^{(h)}(T, I)$, for any constant h, can be adequately estimated in DH, even with limited independence. Accordingly, we will define a witness graph $W^{(h)}(T, I)$ for locations I and corresponding times T. Intuitively, the witness graph ought to contain all vertices (hash keys from D) that could possibly belong to some local dependency tree that comprises h or fewer vertices and has its root located in I.

This witness graph is constructed in a greedy top-down manner, much as the construction of dependency trees. Let $D' = (x'_1, \ldots, x'_{|D'|})$ be the sequence $(D - D_T)$ in reverse order. The witness set will be found by scanning D' to see, essentially, which items might wind up hitting relevant items within their first h probes. We call the locations of relevant items eligible collision points. If an item hits an eligible collision point at its k-th probe, where $k \leq h$, the item is inserted into the witness set and its first $k - 1$ probe locations are inserted into the set of eligible collision points, since these locations must be already occupied by the (real) time the key is actually hashed, if it is to require k probes for insertion. Then the next item is processed. The eligible collision set (i.e., set of eligible collision points) is initialized to I. As this procedure suggests, the witness set is defined without reference to the time constraints T; this simplification can only increase the number of relevant keys and eligible collision points that are found.

We achieve better bounds by including the sum of the number of probes consumed by the sequence of collisions that is responsible for the presence of each eligible collision location. If some item takes k probes to reach an eligible collision location, then only $h-k+1$ probes are available for an item that (at some earlier real insertion time) fills one of the first $k-1$ probe locations in the probe sequence. The following procedure provides a formal construction of the witness graph. The eligible collision set is represented by the family $\mathcal{L}_i(\tau)$, $(i \in [0, h-1])$, where i is an underestimate of the size of the dependency graph which led to the addition of items in $\mathcal{L}_i(\tau)$, and $\tau \in [0, |D'|]$ indicates that the locations in $\mathcal{L}_i(\tau)$ are available for collisions within the first $h-i$ probes of item $x'(\tau+1)$. Initially $\mathcal{L}_0(0) = I$ and $\mathcal{L}_i(0) = \emptyset$ for $i > 0$. After x'_τ is processed and hence $\mathcal{L}_i(\tau)$ determined, $\mathcal{L}_i(\tau+1)$ is initialized to $\mathcal{L}_i(\tau)$ and additional locations might then be inserted into $\mathcal{L}(\tau+1)$, depending on the first h probes of $x'_{\tau+1}$. In particular, if $x'_{\tau+1}$ hits a location in $\mathcal{L}_j(\tau+1)$ on probe i, with $j+i \le h$, then $x'_{\tau+1}$ is inserted into the witness set W and its first $i-1$ probes are inserted into $\mathcal{L}_{j+i-1}(\tau+1)$.

This graph may be viewed as directed and bipartite, with keys and locations as vertices, outgoing edges from a key to locations that correspond to unsuccessful probes, and an incoming edge to a key from the eligible hit location where the key must reside, if it is to belong to some dependency tree of size h or less with root in I.

Because the keys are processed in reverse order, with a greedy interpretation as to their eventual hash location, the procedure will include many elements in the witness set that will turn out to be irrelevant. Some items are assumed to reside in probe locations, for which earlier probe locations will turn out to be empty; for others, the dependency graph will turn out to be to big. On the other hand, these circumstances will only be evident after the structure is completed and all items are inserted.

It is easy to believe that the elements of any actual dependency graph of h or fewer items that is rooted in I are included in the witness set. Actually, we are obliged to state what happens in the unlikely event that several of an item's first h probes hit eligible collision points. The reason that this issue must be addressed is that once two probes are given specific values, the remaining probes may be completely deterministic, in DH.

If an item in $D' - D_T$ incurs multiple collisions, we elect to throw away the "witness", and not record the collisions. This produces a simpler witness graph that overestimates the probability $q^{(h)}(T, I)$ that event $M^{(h)}(T, I)$ occurs, since some witnesses for $M^{(h)}(T, I)$ will not have been included in the graph. On the other hand, we may underestimate $q^{(h)}(T, I)$ as the probability that the vacancy criterion $M^{(h)}(T, I)$ holds and no multiple collisions occur in the witness set.

Definition 16: The witness graph $WG^{(h)}(T, I) = (\mathcal{L}, W, E)$.

Let witness graph $WG^{(h)}(T, I) = (\mathcal{L}, W, E)$ be procedurally specified by the following algorithm.

```
$D' \leftarrow Reverse(D - D_T)$;
$\mathcal{L}_0(0) \leftarrow I_T$;
$W(0) \leftarrow \emptyset$;
for $i \leftarrow 1$ to $h - 1$ do   $\mathcal{L}_i(0) \leftarrow \emptyset$   endfor;
for $\tau \leftarrow 1$ to $|D'|$ do
    for $i \leftarrow 0$ to $h - 1$ do   $\mathcal{L}_i(\tau) \leftarrow \mathcal{L}_i(\tau - 1)$   endfor;
    $W(\tau) \leftarrow W(\tau - 1)$;
    if for exactly one pair $(i, j)$ $i \in [1, h]$, $j \leq h - i : p(x'_\tau, i) \in \mathcal{L}_j(\tau - 1)$ then
        { *A single collision occurs at probe location* $p(x'_\tau, i)$. }
        $\ell \leftarrow p(x'_\tau, i)$;
        $\mathcal{L}_{i+j-1}(\tau) \leftarrow \mathcal{L}_{i+j-1}(\tau) \cup_{k<i} \{p(x'_\tau, k)\}$;
        $W(\tau) \leftarrow W(\tau) \cup \{x'_\tau\}$;
        $E \leftarrow E \cup (\ell, x'_\tau)$ labeled $i$;
        $E \leftarrow E \cup_{k<i} \{(x'_\tau, p(x'_\tau, k))\}$ labeled $k$
    endif
```

```
15.     endfor;
16.     $W \leftarrow W(|D'|)$;
17.     $\mathcal{L} \leftarrow \sum_j \mathcal{L}_j(|D'|)$;
18.     Replace all location labels referencing indices in $L - I$ by pointers to abstract vertices.
```

This procedure enables us to compare the probabilities that a given witness graph structure will occur in UH, DH, and DH_ψ. (See Lemma 5.) To establish an asymptotic equivalence among all three models, it is essential that the actual probe locations apart from those hitting I be absent from the structure (line 18). All that is recorded in the structure is which probes collided with which. To show that the vacancy estimates are almost the same in the three models, we need, in part, estimates to bound, with high probability, the size of a witness graph as a function of the dependency set size $|I|$. Lemma 6 gives a crude (and simple) bound for the expected size of the witness graph, and shows that the witness graph is proportional to $|I|$ with sufficient probability. Lemma 7 gives a better bound on the size of witness sets and thereby establishes a better bound for the independence ψ.

Let $W \subset D'$ be a candidate witness set of k keys with $W = \{x_{i_1}, \ldots, x_{i_k}\}$. Let WG be a candidate labeled witness graph for the pair (T, I) with key vertex set W, and (abstract) eligible location set $\mathcal{L}$, with $\mathcal{L}_i(\tau)$ as defined in the formal procedure. Recall that by construction, any such WG is a forest, when edges are viewed as being undirected; there will be no cycles.

Definition 17.

- Let $\tilde{\mathcal{L}}(\tau) = \sum_{i=0}^{h-1}(h-i)|\mathcal{L}_i(\tau)|$.
- Let $Prob_{UH}^{W}\{WG\}$, $Prob_{DH}^{W}\{WG\}$, $Prob_{DH_\psi}^{W}\{WG\}$ be the respective probability that WG is the actual witness graph for the pair (T, I) in UH, DH and DH_ψ.
- Let $Prob_{UH}^{pure}\{WG\}$, $Prob_{DH}^{pure}\{WG\}$, $Prob_{DH_\psi}^{pure}\{WG\}$ be the respective probabilities that WG is the witness graph for the pair (T, I) and that no vertices where eliminated in its construction due to double hits.

We suppress, for notational simplicity, the implicit dependence of these probabilities on sets

D, T, and I. We shall use the notation WG in two contexts. When WG is selected from a set of candidate witness sets, WG will represent a sample set of keys and a hash structure; here we will compute the probability that WG is the actual witness set that occurs. When WG is not bound as a candidate, it will represent the actual witness set; here the computational issue is the probability that its size $|WG|$ is extremely large. Due to the multiplicity of hashing models, it is convenient to extend Definition 17 as follows.

Definition 18.

- Let $Prob^W\{WG\}$ and $Prob^{pure}\{WG\}$ be used in expressions that hold for each of the three models.
- Let $Prob^W_\bullet\{WG\}$ and $Prob^{pure}_\bullet\{WG\}$ and be used in expressions where $\bullet$ is a free subscript that holds when all $\bullet$-s are simultaneously replaced by UH, DH, or DH_ψ.

Lemma 5. Let WG be a candidate witness graph with $\widetilde{\mathcal{L}} = O(n^{1/2})$. Then

1) $Prob^W_{DH}\{WG\} = Prob^W_{UH}\{WG\}(1 + O(|\mathcal{L}|^2)/n)$.

2) $Prob^W_{DH_\psi}\{WG\} = Prob^W_{DH}\{WG\}(1 + \epsilon e^{-D})$, for $\psi \geq (h|W| + 6\widetilde{\mathcal{L}} + 3|I| + D)$, and some ϵ: $|\epsilon| \leq 1$.

3) $Prob^{pure}_\bullet\{WG\} = Prob^W_\bullet\{WG\}\left(1 + \frac{O(|\mathcal{L}|^2)}{n}\right)$, for UH, DH, and DH_ψ, for $\psi \geq (h|W| + 6\widetilde{\mathcal{L}} + 3|I| + D)$.

Proof: The counting statistics for witness sets is similar to that for dependency graphs, but differs from the latter in two aspects. The simplest change is that witness sets have embedded roots (in locations of I). The other difference is that, unlike the dependency sets of Theorem 1, which are based entirely on local properties within (windows of) k vertices, witness graphs (forests) are global structures selected from the large subsequence $D - D_T$ and locations I. Consequently, the probability that a forest WG is the witness forest in question involves both local hashing properties and the event that the many items not included in WG either do *not hit* any of the eligible hit locations belonging WG or hit the location set more than once. Most importantly, these probabilities turn out to be nearly identical in our three models.

Definition 19.

- Let $Prob_{UH}^{\ell\text{-}hit}\{WG\}$ be the probability that the local set comprising the keys $W \subset D$ hit the prescribed virtual locations in the manner prescribed by WG, and with no additional collisions among the eligible probes of keys in W.

- Let $Prob_{UH}^{not1\text{-}hit}\{WG\}$ be the probability that vertices in $D - D_T - W$ incur either no hit or multiple hits to the eligible location set within the requisite number of probes, as they are processed by the algorithm.

- Let $Prob_{UH}^{no\text{-}hit}\{WG\}$ be the probability that $D - D_T - W$ do not hit the eligible location set at all, within the requisite number of probes.

We extend these definitions to models DH and DH_ψ.

Clearly $Prob_{UH}^{W}\{WG\} = Prob_{UH}^{\ell\text{-}hit}\{WG\} \times Prob_{UH}^{not1\text{-}hit}\{WG\}$. In DH, such a simple formulation is not quite true, since $Prob_{DH}^{\ell\text{-}hit}\{WG\}$ and $Prob_{DH}^{not1\text{-}hit}\{WG\}$ will depend, somewhat, on just which actual locations are used for each possible embedding of WG. Now, $Prob^{\ell\text{-}hit}\{WG\}$ is (in UH, or DH), between $(1/n - O(h|\mathcal{L}|/n^2))^k$ and $(\frac{1}{n-O(1)})^k$, where k is the number of keys belonging to WG. The extra factor of $\frac{1}{n}$ comes from the fact that unlike dependency sets, the roots in witness forests are explicitly embedded in specific locations. In UH, one could evaluate this probability precisely as a function of the sizes $|\mathcal{L}_j(\tau)|$, for a given witness graph. Such an evaluation, however, is unnecessary, since it suffices to show that the computations in the three models are virtually identical. Given a specific embedding of the candidate WG structure, the second factor in $Prob_{UH}^{W}\{WG\}$ is readily written as $Prob_{UH}^{not1\text{-}hit}\{WG\} = Prob\{\bigwedge_{\tau:\ x'_\tau \in D' - W} \neg s(\tau)\}$, where $s(\tau)$ is the event that $x'_\tau \in D'$ experiences a single hit with respect to the embedded sets $\mathcal{L}_0(\tau - 1), \ldots, \mathcal{L}_{h-1}(\tau - 1)$. Let $\sigma(\tau, WG)$, with appropriate subscript UH, DH and DH_ψ, denote $Prob\{s(\tau)\}$. Since, for any specific embedding of WG, the events $s(\tau)$ are mutually independent in both DH and UH, $Prob^{not1\text{-}hit}\{WG\} = \prod_{\tau:\ x'_\tau \in D' - W}(1 - \sigma(\tau, WG))$ in these two models. It is easy to see that for all τ, both $\sigma_{UH}(\tau, WG)$ and $\sigma_{DH}(\tau, WG)$ are, up to factors of $(1 + O(1/n))$, between $\frac{\widetilde{\mathcal{L}}(\tau-1)}{n} = \sum_{i=0}^{h-1}(h - i)\frac{|\mathcal{L}_i(\tau-1)|}{n}$ and $\frac{\widetilde{\mathcal{L}}(\tau-1)}{n} - O(\binom{h}{2})\frac{|\widetilde{\mathcal{L}}(\tau-1)|^2}{n^2}$. Hence

$1 - \sigma_{DH}(\tau, WG) = (1 - \sigma_{UH}(\tau, WG))(1 + \frac{O(|\mathcal{L}(\tau-1)|^2)}{n^2})$, for any embedding of the structure, and any constant h. $Prob^{not1-hit}\{WG\}$ is therefore the same in UH and DH to within a factor of $(1 + O(|\mathcal{L}(\tau)|^2/n^2))^{\tau} = (1 + O(|\mathcal{L}(\tau)|^2/n))$. In DH_{ψ}, the events $s(\tau)$ are only $(\psi - h|W| - 3|I|)$-wise independent, but $\sigma_{DH}(\tau, WG) = \sigma_{DH_{\psi}}(\tau, WG)$, (as long as $\psi > h|W| + \widetilde{\mathcal{L}} + 3|I|$). Lemma A2.2 in the Appendix shows that in this case,

$$|Prob_{DH_{\psi}}^{not1-hit}\{WG\} - Prob_{DH}^{not1-hit}\{WG\}| \leq Prob_{DH}^{not1-hit}\{WG\}e^{-D} \quad \text{if } \psi - h|W| - \widetilde{\mathcal{L}} - 3|I| \geq 5\widetilde{\mathcal{L}} + D,$$

where $\widetilde{\mathcal{L}}$ is used as an overestimate for the expectation $\sum_{\tau \in D'} \sigma_{DH}(\tau, WG)$. Consequently,

1) $Prob_{DH}^{W}\{WG\} = Prob_{UH}^{W}\{WG\}(1 + O(|\mathcal{L}|^2/n))$, and

2) $Prob_{DH_{\psi}}^{W}\{WG\} = Prob_{DH}^{W}\{WG\}(1 + O(e^{-D}))$, for $\psi > h|W| + 6\widetilde{\mathcal{L}} + 3|I| + D$. It is easy to verify 3). Let $Prob_{\bullet}^{W}{}_{D'-x'_{\tau}}\{WG\}$ denote the computation for $Prob_{\bullet}^{W}\{WG\}$ over the set $D' - x'_{\tau}$, where $D' = D - D_T - W$. Then

$$Prob_{\bullet}^{W}\{WG\} - Prob_{\bullet}^{pure}\{WG\} \leq \sum_{\tau} Prob_{\bullet}^{W}{}_{D'-x'_{\tau}}\{WG\}\binom{h}{2}\frac{|WG|^2}{n^2} = Prob_{\bullet}^{W}\{WG\}O(\frac{|WG|^2}{n}),$$

where the factor $\binom{h}{2}\frac{|WG|^2}{n^2}$ is an estimate of the probability that x'_{τ}, has two or more eligible probes into $\mathcal{L}$. ▮

We can now show that $q^{(h)}(T, I)$, the probability that our vacancy criterion holds for locations I, can be successfully approximated in all three models by the probability that these locations are declared empty by our witness graphs. It will follow that the probabilities $q_{UH}^{(h)}(T, I)$, $q_{DH}^{(h)}(T, I)$, $q_{DH_{\psi}}^{(h)}(T, I)$ differ by a factor of at most $\left(1 + \frac{O(|I|^2)}{n}\right)$ in the three models, for an appropriate choice of ψ. Lemmas 6 and 7 both establish this equivalence.

Lemma 6. Let $T = (T_1, \ldots, T_{|T|})$ be a sequence of increasing time stamps with $T_{|T|} = \alpha n$, and let I be an arbitrary sequence of distinct table locations with $|I| = |T|$. In addition, let the following definitions hold.

Definition 20.

- Let the random structure $WG^{(h)}(T, I) = (\mathcal{L}, W, E)$ be the witness graph for D, T, and I.
- Define the random variable $\widetilde{\mathcal{L}}(t) = \sum_{i=0}^{h-1}(h - i)|\mathcal{L}_i(\tau)|$.

- Let $w^{(h)}(T,I)$, with appropriate subscript UH, DH or DH_ψ, be the probability that $WG^{(h)}(T,I)$ contains no witnesses, and hence "declares" each location I_j to be empty at insertion time T_j.

Then for $|I| = o(n^{1/3})$,

1) $w^{(h)}_{DH}(T,I) \le w^{(h)}_{UH}(T,I)\left(1+\frac{O(|I|^2)}{n}\right)$,
$w^{(h)}_{DH_\psi}(T,I) \le w^{(h)}_{DH}(T,I)(1+e^{-D}) + Prob\{|\mathcal{L}| > K\}$, for $\psi \ge 3|I| + 7hK + D$;

2) For $\psi \ge 14(h+2)!(|I|\log\frac{1}{1-\alpha} + \log h + \log n)$,
$q^{(h)}(T,I) = w^{(h)}_{UH}(T,I)\left(1+\frac{O(|I|^2)}{n}\right)$, in UH, DH, and DH_ψ,
and hence witness graphs formalize a good vacancy criterion for all three models.

Proof: It is convenient to analyze the construction of the witness forest as if each collision $p(x'_t, i) \in \mathcal{L}_j(t-1)$ were the outcome of a Bernoulli trial with probability of success $|\mathcal{L}_j(t-1)|/n$. Thus each time step is viewed as contributing $\binom{h}{2}$ (somewhat dependent) Bernoulli trials, of which at most one may result in success. This simplification will have a few insignificant consequences, which we are obliged to acknowledge. The simplest is that the requirement that at most one of the trials be successful can be ignored, since we are interested in establishing upper bounds on the number of successes. The other is that a single (real) probe can hit at most one of the disjoint sets $\mathcal{L}_j$, and modeling the outcome with h different Boolean trials undercounts the probability that at least one success occurs, since the conditional probability that $\mathcal{L}_j$ is probed, given that $\mathcal{L}_0, \ldots, \mathcal{L}_{j-1}$ are not probed can be increased by a factor of about $\frac{n}{n-\sum_{0\le i<j}|\mathcal{L}_j|}$. The simplest resolution of this problem is to include this factor in our model implicitly, by replacing the table size n by the parameter n_1, and recast the probabilities in terms of n_1, which will be adjusted a posteriori. We follow this prescription, although technical arguments can establish that no such a rescaling is actually necessary.

Let $\mathcal{N}(i,t) = \mathrm{E}[|\mathcal{L}_i(t)|]$. We overestimate the probability that $\langle p(x'_t, j-i+1) \in \mathcal{L}_i(t-1)$ *is the only hit of* $x'_t\rangle$ as the outcome of a Bernoulli trial with probability of success equal to $\frac{|\mathcal{L}_i(t-1)|}{n_1}$. We may ignore the time restriction on locations in I as prescribed by the algorithm, and form the system for $\mathcal{N}$ as follows:

$$\mathcal{N}(0,t) = |I|, \text{ and } \mathcal{N}(j,0) = 0,\ j = 1,2,\ldots,h-1;$$
$$\mathcal{N}(j,t) = \mathcal{N}(j,t-1) + \sum_{i<j}(j-i)\mathcal{N}(i,t-1)/n_1. \tag{17}$$

A gross overestimate of $\mathrm{E}[\mathcal{L}_j(\alpha n)]$ (and $\mathcal{N}(j,\alpha n)$) is given by the system:

$$\begin{aligned}&\mathcal{N}_g(0,t) = |I|,\ \mathcal{N}_g(j,0) = 0,\ j = 1,2,\ldots,h-1;\\ &\mathcal{N}_g(j,t) = \mathcal{N}_g(j,t-1) + j\sum_{i<j}\mathcal{N}_g(i,\alpha n)/n_1.\end{aligned} \tag{18}$$

In terms of the random variable $\frac{\mathcal{L}_j(\alpha n)}{j}$, we may define a stochastic dominator $\hat{\mathcal{L}}_j(t) \ge |\mathcal{L}_j(t)|$ with $\mathrm{E}[\hat{\mathcal{L}}_j(t)] = N_g(j,t)$ as follows. Let $\frac{\hat{\mathcal{L}}_j(t)}{j}$ be the outcome of t independent Bernoulli trials with probabilities of success $\sum_{i=0}^{j-1}\frac{\hat{\mathcal{L}}_i(\alpha n)}{n_1}$, where $\hat{\mathcal{L}}_0(t) = I$. We now prove by induction on i that our overestimate of $\sum_{i\le j}\hat{\mathcal{L}}_i(\alpha n)$ (and hence $\sum_{i\le j}|\mathcal{L}_i(\alpha n)|$) is bounded by $B_j = \frac{(j+2)!}{2}B_0$ with probability $1 - je^{-B_0/2}$, for any choice of $B_0 \ge |I|$. It suffices to set, for simplicity and additional overcount, $\alpha n = n_1$. Clearly $\hat{\mathcal{L}}_0(\alpha n) \le B_0$ with probability 1. In general, the probability that $\sum_{i\le j}\hat{\mathcal{L}}_i(\alpha n) > B_j$, subject to the condition that $\sum_{i\le j-1}\hat{\mathcal{L}}_i(\alpha n) \le B_{j-1}$, is bounded by the probability that $\hat{\mathcal{L}}_j(\alpha n) \ge B_j - B_{j-1}$, which can be rewritten as $\frac{\hat{\mathcal{L}}_j(\alpha n)}{j} > (1+1/j)B_{j-1}$, since $B_j = (j+2)B_{j-1}$. $\frac{\hat{\mathcal{L}}_j(\alpha n)}{j}$ is bounded by the number of successes in n_1 Bernoulli trials with probability of success B_{j-1}/n_1. A standard Chernoff-Hoeffding bound for the sum of independent Bernoulli trials, X with expectation $\mathrm{E}[X]$, is $Prob\{X \ge (1+\epsilon)\mathrm{E}[X]\} \le e^{-\epsilon^2\mathrm{E}[X]/3}$, for $0 \le \epsilon \le 1$. This estimate shows that for $t \le n_1$, the probability that $\hat{\mathcal{L}}_j(t)$ exceeds $(1+1/j)B_{j-1}$ is bounded by $e^{-B_{j-1}/3j^2}$, which is at most $e^{-B_0/4}$ for all values $j \ge 1$. Consequently, the probability that $\sum_{i\le j}\hat{\mathcal{L}}_i(\alpha n) > \frac{(j+2)!}{2}B_0$, is bounded by $je^{-B_0/4}$. Let $|\mathcal{L}| = |\mathcal{L}(\alpha n)| = \sum_{i\le h-1}|\mathcal{L}_i(\alpha n)|$. We have shown that

$$Prob\{|\mathcal{L}| > (h+1)!B_0/2\} \le he^{-B_0/4}, \text{for any } B_0 \ge |I|.$$

Let $c_\alpha = -2\log(1-\alpha)$. Choosing $\frac{B_0}{2} = |I|c_\alpha + D$, we see that

$$Prob\{|\mathcal{L}| > (h+1)!(|I|c_\alpha + D)\} \le he^{-(|I|c_\alpha+D)/2} < h(1-\alpha)^{|I|}e^{-D/2}. \tag{19}$$

Furthermore,

$$\begin{aligned}
\sum_{k>c_\alpha|I|(h+1)!} Prob\{|\mathcal{L}|=k\}k^2 &\le \sum_{k>c_\alpha|I|(h+1)!} k^2 h(1-\alpha)^{|I|} e^{-\frac{k-c_\alpha|I|(h+1)!}{2(h+1)!}} \\
&\le h(1-\alpha)^{|I|}(8c_\alpha|I|(h+1)!)^2 \sum_{k>0}\left(\frac{1}{8}+\frac{k}{8c_\alpha|I|(h+1)!}\right)^2 e^{\frac{-k}{2(h+1)!}} \\
&\le h(1-\alpha)^{|I|}(8c_\alpha|I|(h+1)!)^2 \sum_{k>0}\left(1+\frac{k}{8c_\alpha|I|(h+1)!}\right)^2 e^{\frac{-k}{2(h+1)!}} \\
&\le h(1-\alpha)^{|I|}(8c_\alpha|I|(h+1)!)^2 \sum_{k>0} e^{\frac{2k}{8c_\alpha|I|(h+1)!}-\frac{k}{2(h+1)!}} \\
&\le h(1-\alpha)^{|I|}(8c_\alpha|I|(h+1)!)^2 \sum_{k>0} e^{-\frac{k}{4(h+1)!}} \\
&\le h(1-\alpha)^{|I|}(8c_\alpha|I|(h+1)!)^2 \frac{1}{1-e^{-\frac{1}{4(h+1)!}}} \\
&\le h(1-\alpha)^{|I|}(8c_\alpha|I|)^2 5((h+1)!)^3 && (20) \\
&= |I|^2(1-\alpha)^{|I|}O(1). && (21)
\end{aligned}$$

We can now show that the probability that the witness graph $WG^{(h)}(T,I)$ contains no occupancy witnesses for any of the locations is identical in UH and DH, up to a factor of $(1+O(|I|^2/n))$. Evidently, $w^{(h)}(T,I)$ can be expressed as the sum, over all legal candidate witness graphs that declare locations I empty, of the probability that each such graph occurs. Denote by $W_{empty}(T,I)$ the subset of all the witness graphs that declare all locations in I empty at the prescribed times. Let $|WG|$ denote the size of the eligible collision set $|\mathcal{L}|$. Clearly

$$w_{UH}^{(h)}(T,I) = \sum_{WG\in W_{empty}(T,I)} Prob_{UH}^{W}\{WG\}$$

and (22)

$$w_{DH}^{(h)}(T,I) = \sum_{WG\in W_{empty}(T,I)} Prob_{DH}^{W}\{WG\}.$$

We have proven in Lemma 5 that for any witness graph WG, where $|WG| = O(n^{1/2})$,

$$Prob_{DH}^{W}\{WG\} = Prob_{UH}^{W}\{WG\}(1+\frac{O(|WG|^2)}{n})$$

and

$$Prob_{DH_\psi}^{W}\{WG\} \le Prob_{DH}^{W}\{WG\}(1+e^{-D}),\ \text{for}\ \psi \ge (h|W|+6\tilde{\mathcal{L}}_{WG}+3|I|+D).$$

It follows from (19), (21) and (22) that

$$\begin{aligned}
w_{DH}^{(h)}(T,I) \le\ & Prob\{|WG| \ge (c_\alpha|I| + 2\log n)(h+1)!\} \\
& + \sum_{\substack{WG \in W_{empty}(T,I) \\ |WG| < (c_\alpha|I|+2\log n)(h+1)!}} Prob_{UH}^{W}\{WG\}(1 + \frac{O(|WG|^2)}{n}), \\
\le\ & h\frac{(1-\alpha)^{|I|}}{n} + \sum_{\substack{WG \in W_{empty}(T,I) \\ |WG| \le c_\alpha|I|(h+1)!}} Prob_{UH}^{W}\{WG\}\left(1 + \frac{O(c_\alpha|I|(h+1)!)^2}{n}\right) \qquad (23) \\
& + \sum_{\substack{WG \in W_{empty}(T,I) \\ c_\alpha|I|(h+1)! < |WG| \\ |WG| \le (c_\alpha|I|+2\log n)(h+1)!}} Prob_{UH}^{W}\{WG\} \\
& + \sum_{\substack{WG \in W_{empty}(T,I) \\ c_\alpha|I|(h+1)! < |WG| \\ |WG| \le (c_\alpha|I|+2\log n)(h+1)!}} Prob_{UH}^{W}\{WG\}\frac{O(|WG|^2)}{n}, \\
\le\ & \frac{h(1-\alpha)^{|I|}}{n} + w_{UH}^{(h)}(T,I)(1 + O(\frac{(c_\alpha(h+1)!|I|)^2}{n})) + h(1-\alpha)^{|I|}\frac{O(|I|^2((h+1)!)^3c_\alpha^2)}{n},
\end{aligned}$$

where the $O(\frac{(c_\alpha(h+1)!|I|)^2}{n})$ error term comes from (20),

$$\le w_{UH}^{(h)}(T,I)(1 + \frac{O(|I|^2)}{n}),$$

which establishes the first par of 1). The second part: $w_{DH_\psi}^{(h)}(T,I) \le w_{DH}^{(h)}(T,I)(1+e^{-D}) + Prob\{|\mathcal{L}| > K\}$, for $\psi \ge 3|I| + 7hK + D$, follows directly from Lemma 5.

To show that $w^{(h)}(T,I)$ is close to $q^{(h)}(T,I)$ in all three models, we recall that

$$\sum_{WG \in W_{empty}(T,I)} Prob^{pure}\{WG\} \le q^{(h)}(T,I) \le \sum_{WG \in W_{empty}(T,I)} Prob^{W}\{WG\},$$

and that $w^{(h)}(T,I) = \sum_{WG \in W_{empty}(T,I)} Prob^{W}\{WG\}$. Lemma 5 guarantees that each term is close, for witness sets where $\widetilde{\mathcal{L}} = O(n^{1/2})$, and (19) shows that larger sets have a negligible probability of occurrence. Hence $q^{(h)}(T,I) = w^{(h)}(T,I)((1+O(|I|^2)/n)$ in DH and UH. A similarly tight inequality holds for DH_ψ, provided that ψ is sufficiently large. The bounds on ψ are that $\psi \ge (h|W| + 6\widetilde{\mathcal{L}} + 3|I| + \log n)$ from Lemma 5. From (19) and (23), $|\mathcal{L}|$ can be restricted to be no larger than $(c_\alpha|I| + 2\log h + 2\log n)(h+1)!$. Finally, ψ must be large enough that the Chernoff-Hoeffding bound for fully random Bernoulli Trials holds with limited independence. Bound (19) was attained by modeling independent probes as independent Bernoulli trials. From Theorem 5 in

[15], it can be seen that (much more than) sufficient independence is achieved for an independence $\psi - 3|I| - h|W| = \widetilde{\mathcal{L}}$, for the maximum $\widetilde{\mathcal{L}}$ value used in (23). (Alternatively, Lemma A2 can be used, with an increase in ψ by factor of five, and a nominal change in the size of the bound.) Thus, it suffices to set $\psi = 7(h+2)!(c_\alpha|I| + 2\log h + 2\log n)$, where we use the fact that $\widetilde{\mathcal{L}} \leq |h\mathcal{L}|$, and as is easily shown, $|W|$ is unlikely to exceed our bound for $\widetilde{\mathcal{L}}$. The $3|I|$-wise independence for the dependency set is already included in this bound for ψ.

As for the elusive n_1 and the fact that the Bernoulli trials are not completely independent, we see that the probability that some set is hit, conditioned on the event that some other set is not gives a rescaling by at most $\frac{n}{n-\widetilde{\mathcal{L}}}$. As long as $\alpha n \leq n_1 = n - (c_\alpha|I| + 2\log h + 2\log n)(h+1)!$, our results hold as stated. ▮

Recall that a bound for the parameter $|I|$ is achievable in terms of the vacancy estimator and Err_1 in Corollary 4.

The next Lemma shows that ψ need only be proportional to $\log n$ multiplied by a subexponential function of h.

Lemma 7. Given I, let $I_0 = h^3[\ln(1/(1-\alpha)|I| + 2\ln n]$. Let $f(h) = e^{2(h+1/2)^{2/3}}$, and take $\psi \geq 7h^2(3f(h)I_0$.

1) Let $WG^{(h)}(T,I) = (\mathcal{L}, W, E)$, be the witness graph for the pair (T,I), with $T_{|T|} = \alpha n$ and let $\mathcal{N}(j,\alpha n) = \mathrm{E}[\mathcal{L}_j(\alpha n)]$. Then $\mathcal{N}(j,\alpha n) \leq |I|f(j)$ in UH, DH, and DH_ψ.

2) Furthermore,

$$Prob\{|\mathcal{L}| \geq 3hf(h)I_0\} \leq \frac{(1-\alpha)^{|I|}}{n},$$

and

$$Prob\{|W| \geq 3hf(h)I_0\} \leq 2\frac{(1-\alpha)^{|I|}}{n}.$$

3) The vacancy over-estimator $q^{(h)}(T)$ (defined in Definition 14) with respective subscript UH, DH and DH_ψ are equal, up to a factor of $(1 + O(|I|^2)/n)$:

$$q^{(h)}_{DH_\psi}(T,I) = q^{(h)}_{DH}(T,I)\left(1 + \frac{O(1)}{n}\right) = q^{(h)}_{UH}(T,I)\left(1 + \frac{O(|I|^2)}{n}\right).$$

Proof: Equation (17) establishes that a suitable overestimate for the size of the eligible hit set is given by the system:

$$\mathcal{N}(0,0) = |I|, \text{ and } \mathcal{N}(j,0) = 0,\ j = 1,2,\ldots,h-1.$$
$$\mathcal{N}(j,t) = \mathcal{N}(j,t-1) + \sum_{i<j}(j-i)\mathcal{N}(i,t-1)/n_1.$$

It will be convenient to use generating functions and to define a strong inequality $f(x) \preceq g(x)$ to mean that each coefficient in the Taylor expansion of $f(x)$ (about $x = 0$) is positive and at most the value of the corresponding coefficient for g. It will also be convenient, for establishing part 2, to take the following overestimate, where $I_0 > |I|$.

$$\mathcal{N}(j,0) = I_0,\ j = 0,1,2,\ldots,h-1.$$
$$\mathcal{N}(j,t) = \mathcal{N}(j,t-1) + \sum_{i<j}(j-i)\mathcal{N}(i,t-1)/n,$$

and to extend the definition of $\mathcal{N}$ to larger values of j by setting

$$\begin{aligned} &\mathcal{N}(j,0) = I_0,\ j = 0,1,2,\ldots. \\ &\mathcal{N}(j,t) = \mathcal{N}(j,t-1) + \sum_{i<j}(j-i)\mathcal{N}(i,t-1)/n. \end{aligned} \tag{24}$$

This latter modification does not even affect the values of $\mathcal{N}(j,0)$, for $j < h$.

Let $\nu(t,x)$ be the generating function $\nu(t,x) = \sum_j \mathcal{N}(j,t)x^j$. The solution for ν is immediate. The initial condition is $\nu(0,x) = \sum_j I_0 x^j = \frac{I_0}{1-x}$, and the recurrence equation becomes

$$\begin{aligned} \nu(t,x) &= \nu(t-1,x) + \sum_j \sum_{i<j}(j-i)\mathcal{N}(i,t-1)x^j/n_1 \\ &= \nu(t-1,x) + \frac{1}{n_1}\sum_j \mathcal{N}(j,t-1)x^j \sum_k kx^k \\ &= \nu(t-1,x) + \frac{\nu(t-1,x)}{n_1}(\sum_k kx^k) = \nu(t-1,x)\left(1 + \frac{1}{n_1}\frac{x}{(1-x)^2}\right). \end{aligned}$$

Hence

$$\nu(t,x) = (1 + \frac{x}{n_1(1-x)^2})^t \frac{I}{1-x}.$$

Now, $1 + \frac{x}{n_1(1-x)^2} \preceq e^{\overline{\frac{x}{n_1(1-x)^2}}}$, (since $e^{\overline{\frac{x}{(1-x)^2}}} = 1 + \frac{x}{(1-x)^2} + \sum_{j>1} x^j \frac{1}{j!(1-x)^{2j}}$), whence for all $t \geq 0$, $(1 + \frac{x}{n_1(1-x)^2})^t \preceq e^{\overline{\frac{tx}{n_1(1-x)^2}}}$. Thus $\nu(t,x) \preceq e^{\overline{\frac{tx}{n_1(1-x)^2}}} \frac{I}{1-x}$, and $\mathcal{N}(j,t)$ is bounded by the jth term in the Taylor expansion of $e^{\overline{\frac{tx}{n_1(1-x)^2}}} \frac{I}{1-x}$. Evidently, $\nu(t,z)$ is defined for all $|z| < 1$. Consequently, its

Taylor coefficients grow subexponentially. Indeed, any analytic function, which has positive Taylor coefficients and has an infinite subsequence of coefficients a_{λ_i} as large as $(\frac{1}{\rho})^{\lambda_i}$, for any fixed $\rho < 1$, cannot be bounded as $x \to \rho$, since the summation will have an infinite number of terms that become at least as large as 1. Similarly, if the Taylor coefficients of f were bounded by some polynomial $|a_j| < cj^k$ for fixed k and c, then the $k+2$ fold iterated integral of f would have coefficients of size $O(a_j/j^{k+2}) = O(j^{-2})$, which would render the integrated function convergent on $|z| = 1$. It follows that the coefficients of $e^{\frac{tz}{n_1(1-z)^2}} \frac{I}{1-z}$ must be superpolynomial since the function has an essential singularity (i.e. poles $(\frac{1}{z-1})^d$ for unbounded degree d) at $z = 1$. A more precise estimation of the coefficients, with $t = \alpha n = \alpha_1 n_1$ follows.

$$\frac{\nu(\alpha_1 n_1, x)}{|I|} \leq \sum_j \frac{\alpha_1^j x^j}{j!(1-x)^{2j+1}} = \sum_j \frac{x^j \alpha_1^j}{j!} \sum_{k \geq 2j} \binom{k}{2j} x^{k-2j}$$
$$= \sum_j \sum_{k \geq 2j} \frac{x^{k-j} \alpha_1^j}{j!} \binom{k}{2j} = \sum_j x^j \left(\sum_{\ell=0}^{j} \binom{\ell+j}{2\ell} \frac{\alpha_1^\ell}{\ell!} \right) .$$

Consequently, $\frac{\mathcal{N}(j,\alpha_1 n)}{|I|} \leq \sum_{\ell=0}^{j} \binom{\ell+j}{2\ell} \frac{\alpha_1^\ell}{\ell!}$. Now, $\binom{\ell+j}{2\ell}\frac{1}{\ell!} \leq (j+1/2)^{2\ell} \binom{3\ell}{\ell} \frac{1}{3\ell!}$, and $\binom{3\ell}{\ell} \leq \frac{3^{3\ell}}{2^{2\ell}}$, so the summation is bounded by $\sum_{\ell=0}^{j} \frac{\left(\frac{3}{2^{2/3}}(j+1/2)^{2/3}\alpha_1^{1/3}\right)^{3\ell}}{(3\ell)!} < e^{2\alpha_1^{\frac{1}{3}}(j+1/2)^{2/3}}$. Setting $\alpha_1 = 1$ establishes 1).

Given a dependency set size $|I|$, let $\mathcal{N}(j,t)$ be defined by (24) with $I_0 = a|I| + b$, where a and b will be specified later. We now show that for suitable a and b, $|\mathcal{L}_j(t)|$ is with very high probability no larger than $3\mathcal{N}(j,t)$.

Formally, we analyze the following modified process: if at any time τ, $|\mathcal{L}_j(\tau)| > (1+1/h)^j \mathcal{N}(j,\tau)$, then the process is aborted, and failure declared. The probabilistic recurrence analogous to (17) is given below, where $\mathcal{X}\langle event \rangle$ denotes the indicator function for the *event.*

$$|\mathcal{L}_0(\tau)| = |I|, \text{ and } |\mathcal{L}_j(\tau)| = 0,\ j = 1, 2, \ldots, h-1.$$
$$|\mathcal{L}_j(\tau)| = |\mathcal{L}_j(\tau-1)| + \sum_{i<j} (j-i)\mathcal{X}\langle p(y_\tau, j-i+1) \in \mathcal{L}_i(\tau-1)\rangle.$$

Expanding the recursion gives:

$$|\mathcal{L}_j(\tau)| = \sum_{i=0}^{j-1} (j-i) [\sum_{\tau' \leq \tau} \mathcal{X}\langle p(y_{\tau'}, j-i+1) \in \mathcal{L}_i(\tau'-1)\rangle], \text{ for } j > 0 . \qquad (25)$$

where the event $\langle p(y_{\tau'}, j-i+1) \in \mathcal{L}_i(\tau'-1)\rangle$ requires that no other probe of $y_{\tau'}$ within the first h hit an eligible collision location.

A similar expansion for $\mathcal{N}$ gives:

$$\mathcal{N}(i,\tau) = \sum_{j=0}^{i-1}(i-j)\sum_{\tau'<\tau}\mathcal{N}(j,\tau')/n_1 + I_0. \tag{26}$$

We are now ready to show that in DH, with probability $1-(\alpha n(h-1))e^{-\frac{I_0}{h^3}}$ or more, $|\mathcal{L}_j(\tau)| < (1+1/h)^j\mathcal{N}(j,\tau)$ for $j=0,1,\ldots,h-1$ and $t=1,2,\ldots,\alpha n$. The bound is clearly true for $j=0$. The method of proof is to compute the probability that the bound fails to hold, for each pair (j,τ), given that it holds for all smaller (i,s), $i<j$, $s<\tau$.

By construction, $\sum_{\tau'\le\tau}\mathcal{X}\langle p(y_{\tau'}, i-j+1)\in\mathcal{L}_j(\tau'-1)\rangle$, is statistically dominated by the sum of independent Bernoulli trials $X(\tau',j)$ with probability of success equal to $\frac{|\mathcal{L}_j(\tau'-1)|}{n_1}$, for $\tau'=1,\ldots,\tau$. Let $X_j(t)=\sum_{\tau=1}^{t}X(\tau,j)$. By assumption,

$$E(X_j(t)) \le [\sum_{\tau<t}(1+1/h)^j\frac{\mathcal{N}(j,\tau)}{n_1}].$$

We now use the following type of Chernoff-Hoeffding bound, (proven in Lemma A1 in the Appendix), to bound our Bernoulli process:

$$Prob\Big\{X_j(t)\ge(1+1/h)E[X_j]+C\Big\}\le e^{-\frac{5C}{4h}}.$$

Let $C=2I_0/h^2$, so that with probability exceeding $1-e^{-\frac{2I_0}{h^3}}$, an individual $X_j(t)$, $(0\le j\le i-1)$, is bounded by $\Big[\sum_{\tau\le t}(1+1/h)^{j+1}\frac{\mathcal{N}(j,\tau-1)}{n_1}+\frac{2I_0}{h^2}\Big]$, if the earlier $X_i(\tau)$-s satisfy their respective bounds for $\tau\le t-1$. According to the definitions of X_i and $\mathcal{L}_i$, for $i>0$,

$$|\mathcal{L}_i(t)|\le\sum_{j=0}^{i-1}(i-j)X_j(t),$$

whence,

$$\begin{aligned}|\mathcal{L}_i(t)| &\le \sum_{j=0}^{i-1}(i-j)\left[\sum_{\tau\le t}(1+1/h)^{j+1}\frac{\mathcal{N}(j,\tau-1)}{n_1}+\frac{2I_0}{h^2}\right]\\ &\le I_0+(1+1/h)^i\sum_{j=0}^{i-1}(i-j)\left[\sum_{\tau\le t}\frac{\mathcal{N}(j,\tau-1)}{n_1}\right],\end{aligned}$$

which by (26) is

$$\leq (1+1/h)^i \mathcal{N}(i,t).$$

Hence $\forall \tau \leq t$, $i < h$,

$$|\mathcal{L}_i(\tau)| \leq (1+\frac{1}{h})^i \mathcal{N}(i,\tau), \text{ with probability } 1-(ti)e^{-\frac{2I_0}{h^3}}.$$

Now let $I_0 = \frac{h^3}{2}[\ln(1/(1-\alpha_1))|I|+3\ln n]$, so that $(hn)e^{-\frac{2I_0}{h^3}} \leq \frac{(1-\alpha_1)^{|I|}}{n}$. Upon substituting, we see that in DH (and UH), the size of the eligible collision set $|\mathcal{L}(\alpha_1 n)|$ is bounded by $3h\mathcal{N}(h-1, \alpha_1 n)$, with probability $1-\frac{(1-\alpha)^{|I|}}{n}$.

The size of the witness set W can be bounded similarly. Let $W_j(\tau)$ be the witness set at time τ containing elements that were due to a hit to $\mathcal{L}_j$:

$$|W_j(\tau)| \leq \sum_{\tau' \leq \tau} \mathcal{X}\langle p(y_{\tau'}, i) \in \mathcal{L}_j(\tau'-1) \text{ for some probe number } i \in [1, h-j]\,\rangle.$$

If we are given upper bounds $R_j(\tau'-1)$ for the size of the $\mathcal{L}_j(\tau'-1)$, then $|W_j(\tau)|$ is bounded by the sum of τ independent Bernoulli trials $X_{\tau'}$, with probabilities of success $p_{\tau'} \leq (h-j)R_j(\tau'-1)/n_1$. We have just shown that $\sum_{\tau' \leq \tau}(h-j)|\mathcal{L}_j(\tau'-1)|/n_1$ is with probability $1-(th)e^{-\frac{2I_0}{h^3}}$ bounded by $\sum_{\tau' \leq \tau}(1+1/h)^j(h-j)|\mathcal{N}_j(\tau'-1)|/n_1$. The Chernoff-Hoeffding bound from Lemma A2 and the bound for $\mathcal{L}_j(\tau)$ gives:

$$Prob\{|W_j(\tau)|] > (1+1/h)\sum_{\tau' \leq \tau}(1+1/h)^j(h-j)|\mathcal{N}_j(\tau'-1)|/n_1 + \frac{2I_0}{h^2}\} \leq e^{-\frac{2I_0}{h^3}}.$$

Hence $W(\tau)$ is with probability $1-2h\tau e^{-\frac{2I_0}{h^3}} \geq 1-2\frac{(1-\alpha)^{|I|}}{n}$ bounded by

$$\sum_{j=0}^{h-1}\left(\sum_{\tau' \leq \tau}(1+1/h)^{j+1}(h-j)|\mathcal{N}_j(\tau'-1)|/n_1 + 2\frac{I_0}{h^2}\right) \leq 3\mathcal{N}(h,\tau).$$

We again take $\tau = n_1$ and establish the bound for the witness set as stated.

Part 3) follows from Lemma 6 and the bounds for $|W|$ and $|\mathcal{L}|$. ▮

The only remaining step is to prove that witness graphs give multiplicative vacancy overestimators.

Theorem 3.

1) $q_{UH}^{(h)}(t) \equiv q_{UH}^{(h)}(t,1)$ is a multiplicative vacancy overestimator for $M_{UH}^{(h)}(T,I)$. Hence, for any fixed $\alpha < 1$, fixed h, and $T \subset [1, \alpha n]$ with $|T| = O(\log n)$,

$$q_{UH}^{(h)}(T,I) = (1 + \frac{O(|T|^2)}{n}) \prod_{i \le |T|} q_{UH}^{(h)}(T_i).$$

2) In DH and DH_ψ, for $\psi \ge 21h^5 e^{2(h+1/2)^{2/3}}$, we have the multiplicative vacancy overestimator

$$q_{DH_\psi}^{(h)}(t) = 1 - \frac{t}{n} + \frac{(2\frac{t}{n} - (\frac{t}{n})^2)^{h+1}}{\sqrt{h+1}(1-\frac{t}{n})^2} + \frac{O(1)}{n}.$$

Proof: Lemmas 5, 6 and 7 establish that

$$\begin{aligned} q_{UH}^{(h)}(T,I) &= \left(1 + \frac{O(|I|^2)}{n}\right) \sum_{WG \in W_{empty}(T,I)} Prob_{UH}^{pure}\{WG\} \\ &= \left(1 + \frac{O(|I|^2)}{n}\right) \sum_{WG \in W_{empty}(T,I)} Prob_{UH}^{\ell-hit}\{WG\} \times Prob_{UH}^{no-hit}\{WG\}, \end{aligned}$$

where $W_{empty}(T,I)$ denotes the set of all graphs of $O(I + \log n)$ keys that could be the witness graph for the pair (T,I), and which declare all I locations empty. $Prob_{UH}^{\ell-hit}\{WG\}$ is the probability that vertices included in the witness graphs behave as prescribed and $Prob_{UH}^{no-hit}\{WG\}$ is the probability that no other vertex has an eligible hit.

We may partition each forest WG into the $|I|$ trees $WG_1, \ldots, WG_{|I|}$, rooted at the respective locations $I_1, \ldots, I_{|I|}$. It is easy to see that $Prob_{UH}^{\ell-hit}\{WG\} = (1 + O(\frac{|WG|^2}{n})) \prod_{i=1}^{|I|} Prob_{UH}^{\ell-hit}\{WG_i\}$, where the factor $(1 + O(\frac{|WG|^2}{n}))$ is needed to account for the conditioning needed to ensure that the trees have disjoint embeddings. Lemma 5 remarks that in UH, $Prob_{UH}^{no-hit}\{WG\}$ can be expressed as $\prod_{\tau:\ x'_\tau \in D' - W} (1 - \sigma^*(\tau, WG))$, and $Prob_{UH}^{no-hit}\{WG_i\} = \prod_{\tau:\ x'_\tau \in D' - W^i} (1 - \sigma^*(\tau, WG_i))$. It is not hard to verify that for any $x_\tau \in D' - W^i$, $\sigma^*(\tau, WG_i) = (1 + \frac{O(|WG|)}{n})\sigma^{**}(\tau, WG_i)$, where $\sigma^{**}(\tau, WG_i)$ is the conditional probability that x_τ has an eligible probe hitting WG_i given that its eligible probes do not hit the relevant portions of $WG_1, WG_2, \ldots, WG_{i-1}$. Consequently, $1 - \sigma^*(\tau, WG_i) =$

$(1+\frac{O(|WG||WG_i|)}{n^2})(1-\sigma^{**}(\tau, WG_i))$. Multiplying over τ and i gives:

$$
\begin{aligned}
q_{UH}^{(h)}(T,I) = &\sum_{WG \in W_{empty}(T,I)} \\
&\quad (1+\frac{O(|WG|^2)}{n}) Prob_{UH}^{\ell\text{-}hit}\{WG\} \prod_{i=1}^{|I|} \prod_{\tau:\ x'_\tau \in D'-W} (1+\frac{O(|WG||WG_i|)}{n^2})(1-\sigma^{*}(\tau, WG_i)) \\
= &\sum_{WG \in W_{empty}(T,I)} \\
&\quad (1+\frac{O(|WG|^2)}{n}) \prod_{i=1}^{|I|} Prob_{UH}^{\ell\text{-}hit}\{WG_i\}(1+\frac{O(|WG||WG_i|)}{n}) \prod_{\tau:\ x'_\tau \in D'-W} (1-\sigma^{*}(\tau, WG_i)) \\
= &\sum_{WG \in W_{empty}(T,I)} (1+\frac{O(|WG|^2)}{n}) \prod_{i=1}^{|I|} Prob_{UH}^{\ell\text{-}hit}\{WG_i\} \prod_{\tau:\ x'_\tau \in D'-W^i} (1-\sigma^{*}(\tau, WG_i)) \\
\leq &\prod_{i=1}^{|I|} \sum_{WG_i \in W_{empty}(T_i,I_i)} (1+\frac{O(|WG||WG_i|)}{n}) Prob_{UH}^{\ell\text{-}hit}\{WG_i\} \prod_{\tau:\ x'_\tau \in D'-W^i} (1-\sigma^{*}(\tau, WG_i)).
\end{aligned}
$$

Evaluating each outer factor with size limits as in (23) gives

$$q_{UH}^{(h)}(T,I) \leq \prod_{i=1}^{|I|} (1+\frac{O(|I|)}{n}) w_{UH}^{(h)}(T_i - |I|, I_i), \tag{27}$$

where we take $w_{UH}^{(h)}(t,\ell)$ to be 1, for $t<1$, and subtract $|I|$ from T_i in (27) because D' comprises the elements $D-D_T$.

Appealing to Lemma 6.2 gives:

$$q_{UH}^{(h)}(T,I) \leq (1+\frac{O(|I|^2)}{n}) \prod_{i=1}^{|I|} q_{UH}^{(h)}(T_i - |I|),$$

whence a final simplification shows that

$$q_{UH}^{(h)}(T,I) \leq (1+\frac{O(|I|^2)}{n}) \prod_{i=1}^{|I|} q_{UH}^{(h)}(T_i), \tag{28}$$

which establishes 1).

Claim 2) is a direct consequence of 1), Lemma 4, and Lemma 7.

To be precise, (28) follows for our specific function $q_{UH}^{(h)}$. Modestly annoying combinatorial arguments would be needed to establish (28) in full generality; we forbear.

Part 3 of Lemma 7 ensures that $q_{DH}^{(h)}$ and $q_{DH_\psi}^{(h)}$ inherit a multiplicative formulation comparable to that for $q_{UH}^{(h)}$. ▮

4. Conclusions

We have shown that in double hashing, a universal family of $\log n$-wise independent hash functions can give nearly optimal performance for any fixed load bounded below 1.

These results comprise a significant step toward understanding why extremely simple functions seem to perform so well when used to double hash arbitrary values into a partially filled table. Indeed, it is quite conceivable that real data, when hashed by such functions, might yield sequences that exhibit $O(\log n)$-wise independence.

Our proof technique analyzed local and global hashing interactions separately, and used analytic tools to measure complicated but weakly correlated events in terms of simpler independent processes. Surely these methods can be applied to other probabilistic processes that exhibit weak correlations and that might be supported only by a source of limited randomness.

5. Appendix

This section contains two technical Lemmas, which can simplify large deviation calculations in cases of full and limited independence. Lemma A.2 is a special case of Theorem 6 in [15].

Lemma A1. Let $X = \sum_{i=1}^{n} X_i$ be the sum of n mutually independent Bernoulli trials $X_1, \ldots, X_n$, where $Prob\{X_i = 1\} = p_i$. Then

$$\text{for any } C > 0, \text{ and } 0 \le \epsilon \le 1,\ Prob\{X \ge (1+\epsilon)\mathrm{E}[X] + C\} \le e^{-1.25\epsilon C}.$$

Proof: Let $p = \frac{\mathrm{E}[X]}{n}$. According to Hoeffding, [Ho-63]:

$$Prob\{X \ge (1+\delta)\mathrm{E}[X]\} \le \left(\frac{1}{1+\delta}\right)^{(1+\delta)\mathrm{E}[X]} \left(\frac{1-p}{1-(1+\delta)p}\right)^{n-(1+\delta)\mathrm{E}[X]} \le \left(\frac{1}{1+\delta}\right)^{(1+\delta)\mathrm{E}[X]} e^{\delta \mathrm{E}[X]}.$$

Let $C = (\delta - \epsilon)\mathrm{E}[X]$. It suffices to show that for any $\delta > 0$, and $0 \le \epsilon \le 1$,

$$\left(\frac{1}{1+\delta}\right)^{(1+\delta)\mathrm{E}[X]} e^{\delta \mathrm{E}[X]} \le e^{-1.25(\epsilon)(\delta-\epsilon)\mathrm{E}[X]}.$$

(For $\delta \le 1$, a simple derivation gives the slightly better bound where the 1.25, is replaced by $\frac{3}{2}$.)

We therefore need only show that for all $0 \le \epsilon \le 1$,

$$\left(\frac{1}{1+\delta}\right)^{(1+\delta)} e^{\delta} \le e^{-1.25\epsilon(\delta-\epsilon)}.$$

Let $g(\delta,\epsilon) = \left(\frac{1}{1+\delta}\right)^{(1+\delta)} e^{\delta+1.25\epsilon(\delta-\epsilon)}$. For any fixed ϵ, $g(\delta,\epsilon)$ attains its maximum at $1+\delta = e^{1.25\epsilon}$. Therefore $\log g(\delta,\epsilon) \le \log g(e^{1.25\epsilon}-1,\epsilon) = e^{1.25\epsilon} - 1 - 1.25\epsilon - 1.25\epsilon^2$. For $\epsilon > 0$, the expression $e^{1.25\epsilon} - 1 - 1.25\epsilon - 1.25\epsilon^2$ first decreases and then increases since it is initially decreasing and its derivative is the sum of a linear function and a function with a rapidly increasing derivative. Hence $f(\epsilon) = g(e^{1.25\epsilon}-1,\epsilon)$ first decreases and then increases. It follows that $f(\epsilon) \le \max\{f(0), f(1)\}$, for $\epsilon \in [0,1]$. Since $\log f(0) = 0$, and calculation shows that $\log f(1) < -.009$, we see that $g(\delta,\epsilon) < 1$, for all δ, and $\epsilon \in [0,1]$. ▮

Lemma A2. Let $X_1, X_2, \ldots, X_n$ and $Y_1, Y_2, \ldots, Y_n$ be Bernoulli trials with probabilities of success $\mathrm{E}[X_i] = \mathrm{E}[Y_i] = p_i$. Let $X = \sum_{i=1}^n X_i$. Suppose that the Y_i-s are mutually independent, and that the X_i-s are fully ψ-wise independent. Let $I = \{i_1, i_2, \ldots, i_k\}$, and let $\widehat{I} = [1,n] - I$. Let $p(I) = Prob\{\left(\wedge_{j\in I}(Y_{i_j} = 1)\right) \wedge \left(\wedge_{j\in \widehat{I}}(Y_{i_j} = 0)\right)\}$, and let $p_\psi(I) = Prob\{\left(\wedge_{j\in I}(X_{i_j} = 1)\right) \wedge \left(\wedge_{j\in \widehat{I}}(X_{i_j} = 0)\right)\}$, so that the subscript ψ indicates that the event is with respect to the fully ψ-wise independent trials $X_1, X_2, \ldots, X_n$.

(1) For $\psi \ge D + k + \epsilon\mathrm{E}[X] - \log\left(\prod_{i\in\widehat{I}}(1-p_i)\right)$ and some ϵ where $|\epsilon| \le 1$,

$$p_\psi(I) = p(I)(1+\epsilon e^{-D})$$

(2) If $\forall i$: $p_i \le 1/2$, then for $\psi \ge D + k + 5\mathrm{E}[X]$ and some ϵ where $|\epsilon| \le 1$,

$$p_\psi(I) = p(I)(1+\epsilon e^{-D}).$$

Proof: The proof of Lemma A2 is a special case of Theorem 6 in [15]. It is given here for completeness.

We may use standard inclusion-exclusion to estimate the probability $p_\psi(I)$ as follows.

$$\begin{aligned} p_\psi(I) &= Prob_\psi\{\left(\bigwedge_{j\in I}(X_j = 1)\right) \wedge \left(\bigwedge_{\ell\notin I}(X_\ell = 0)\right)\} \\ &= Prob_\psi\{\bigwedge_{j\in I}(X_j = 1)\}\sum_{\ell=0}^{n}\sum_{i_{k+1}<\ldots<i_{k+\ell}\notin I}(-1)^\ell Prob_\psi\{\bigwedge_{j\in i_{k+1}<\ldots<i_{k+\ell}}(X_j = 1)\}. \end{aligned}$$

Truncating the outer summation at $\ell = \psi - k$ introduces an error that is bounded by the last term of the truncated sum. Let $p_\psi^T(k)$ and $p^T(k)$ denote these truncated sums, in the respective cases of ψ-wise and full independence. Since the first $\psi - k$ terms in the outer summation are the same for both fully and ψ-wise independent random variables, $p^T_\psi(k) = p^T(k)$. Furthermore, $Prob_\psi\{\bigwedge_{j\in\{i_1,\ldots,i_{k+l}\}}(X_j = 1)\} = \prod_{j=1}^{k+l} p_{i_j}$. Hence

$$p_\psi(k) = p^T(k) - (-1)^{\psi-k}\delta_\psi[\prod_{j\in I} p_{i_j}] \sum_{\substack{i_1<\ldots<i_{\psi-k}\\ i_j\notin I}} \prod_{j=1}^{\psi-k} p_{i_j},$$

for some $\delta_\psi \in [0,1]$, and an identical inequality holds without the ψ subscripts. Hence

$$|p_\psi(I) - p(I)| \le [\prod_{j\in I} p_{i_j}] \sum_{i_1<\ldots<i_{\psi-k}} \prod_{j=1}^{\psi-k} p_{i_j}.$$

$\sum_{i_1<\ldots<i_{\psi-k}} \prod_{j=1}^{\psi-k} p_{i_j}$ is maximized when all p_{i_j} are equal and therefore the error $|p_\psi(k)-p(k)|$ is bounded by

$$[\prod_{j\in I} p_{i_j}]\binom{n}{\psi-k}\left(\frac{p_1+p_2+\ldots+p_n}{n}\right)^{\psi-k} \le [\prod_{j\in I} p_{i_j}](\mathrm{E}[X])^{\psi-k}/(\psi-k)!,$$

To get multiplicative error bounds, we need that $(\mathrm{E}[X])^{\psi-k}/(\psi-k)! \le e^{-D}\prod_{j\notin I}(1-p_j)$. Setting $\psi - k = e\mathrm{E}[X] - \log\left(\prod_{j\notin I}(1-p_j)\right) + D$, gives:

$$(\mathrm{E}[X])^{\psi-k}/(\psi-k)! \le \left(\frac{e\mathrm{E}[X]}{\psi-k}\right)^{\psi-k} \le \left(1 + \frac{\log\left(\prod_{j\notin I}(1-p_j)\right) + D}{\psi-k}\right)^{\psi-k} \le e^{-D}\prod_{j\notin I}(1-p_j).$$

This proves 1. The second inequality follows immediately by observing that if, say, $\forall i$: $p_i \le \frac{1}{2}$, then

$$-\log\left(\prod_{j\notin I}(1-p_j)\right) = \sum_{j\notin I}\sum_{k>0}\frac{p_j^k}{k} \le \sum_{j\notin I} 2p_j \sum_{k>0}\frac{(1/2)^k}{k} = -\sum_j 2p_j \log\frac{1}{2} < (1.4)\mathrm{E}[X]. \quad \blacksquare$$

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[2] M. Ajtai, J. Komlós, E. Szemerédi. There Is No Fast Single Hashing Algorithm, *IPL*, 7,6, 1978, pp. 270–273.

[3] B. Bollobás, A.Z. Broder, and I. Simon. The cost distribution of clustering in random probing, to appear, *JACM*.

[4] R Brent. Reducing the Retrieval Time of Scatter Storage Techniques, *CACM*, **16**(2), 1973, pp. 105–109.

[5] H. Chernoff. A measure of asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations, *Ann. Math. Statist.*, 23 (1952), pp. 493-507.

[6] J.L. Carter and M.N. Wegman. Universal Classes of Hash Functions, *JCSS*, 18, 1979, pp. 143–154.

[7] D.H. Greene and D.E. Knuth. *Mathematics for the analysis of algorithms*, Birkhauser, 1990.

[8] L. Guibas and E. Szemerédi. The Analysis of Double Hashing, *JCSS*, 16, 1978, pp. 226–274.

[9] W. Hoeffding. On the Distribution of the Number of Successes in Independent Trials, *Ann. Math. Statist.*, 27 (1956), pp. 713-721.

[10] D.E. Knuth. *The Art of Computer Programming, Vol. 3: Sorting and Searching*, Addison-Wesley, Reading, Mass. 1973.

[11] G. Lueker and M. Molodowitch. More Analysis of Double Hashing, *20*th *STOC*, May, 1988, pp. 354–359.

[12] K. Mehlhorn. On the Program size of Perfect and Universal Hash functions, *Proc. 23rd Ann. Symp. on Foundations of Computer Science*, 1982, pp. 170–175.

[13] K. Mehlhorn. *Data Structures and Algorithms 1: Sorting and Searching*, Springer-Verlag, Berlin Heidelberg, 1984.

[14] J.P. Schmidt and A. Siegel. On aspects of universality and performance for closed hashing, *21*st *STOC*, May, 1989, pp. 355–366.

[15] J.P. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-Hoeffding Bounds for Applications with Limited Independence. *Proc. 4th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 1993, 331–340. To appear *SIAM J. Discrete Math.*

[16] A. Siegel. On universal classes of fast hash functions, their time-space tradeoff, and their applications, *Proc. 30th Ann. Symp. on Foundations of Computer Science*, Oct., 1989, pp. 20–25.

[17] A. Siegel and J.P. Schmidt. Closed hashing is computable and optimally randomizable with universal hash functions, submitted.

[18] J.D. Ullman. A Note on the Efficiency of Hash Functions, *JACM*, Vol 19, 1972, pp. 569–575.

[19] M.N. Wegman and J.L. Carter. New Hash Functions and Their Use in Authentication and Set Equality, Journal of Comp. Syst. Sci. 22, 1981, pp. 265–279.

[20] A.C. Yao. Uniform Hashing Is Optimal, *JACM*, Vol 32, No. 3, July, 1985, pp. 687–693.